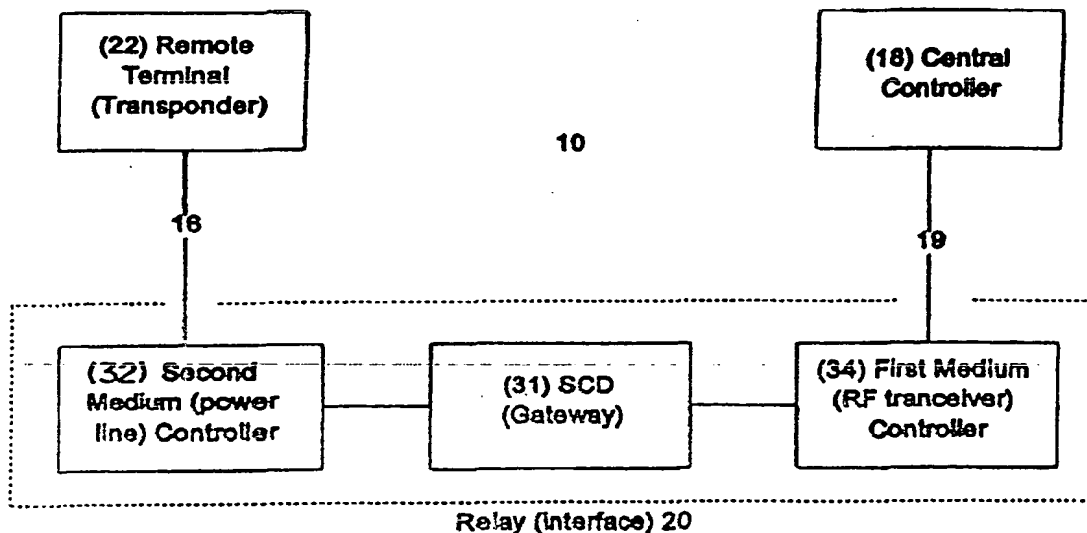




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : G06F 13/36		A1	(11) International Publication Number: WO 99/09488
			(43) International Publication Date: 25 February 1999 (25.02.99)
(21) International Application Number: PCT/US98/16734		(81) Designated States: CA, CN, KR, US, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).	
(22) International Filing Date: 13 August 1998 (13.08.98)			
(30) Priority Data: 08/911,809 15 August 1997 (15.08.97) US		Published With international search report.	
(71) Applicant (for all designated States except US): MISSISSIPPI POWER COMPANY [US/US]; 2992 West Beach Boulevard, Gulfport, MS 39501 (US).			
(72) Inventors; and (75) Inventors/Applicants (for US only): SEAL, Robert, Bryan [US/US]; 271 Oakwood Drive, Meridian, MS 39305 (US). LABOE, Steven, P. [US/US]; 3919 Peninsula Drive, Lauderdale, MS 39335 (US).			
(74) Agent: GABLE, R., Lewis; Cowan, Liebowitz & Latman, P.C., 1133 Avenue of the Americas, New York, NY 10036-6799 (US).			

(54) Title: SCALEABLE COMMUNICATIONS DEVICE



(57) Abstract

A scaleable or adaptive communications device (31) or gateway for serving as a bridge (20) between a local area network (LAN) (30) and a wide area network (WAN) (23). In one preferred embodiment of the invention the LAN takes the form of a segment of a power line carrier (PLC) (32) as defined by transformers on either end. A communications device (31) for serving as a gateway between an LAN (30) and a WAN (23) which is capable of being enhanced to expand the gateway to undertake higher level functions, including computing analyzing and transmitting collected data to achieve a variety of preprogrammed objectives.

BEST AVAILABLE COPY

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

SCALEABLE COMMUNICATIONS DEVICE

BACKGROUND OF THE INVENTION

5 The present invention relates generally to the field of electronic data communication devices and more particularly to the field of such devices which are "scaleable" or adaptive to allow an increase in their functionality by the addition of a number of elements.

10 The Applicant is aware of the following prior art references which are generally related to the invention here:

15 Mathias et al., U.S. Patent No. 5,446,897 relates to a method of assigning addresses to devices coupled to a network and maintaining an address list of the coupled devices. As shown in FIG. 2, the connectable devices take the form of work stations 38, 68, servers 46, 70 and computers 42. However, Mathias does not appear to teach that these devices function or otherwise cooperate with each other as a gateway, i.e., there is no disclosure

20 that there is a system for bidirectionally communicating data between a central processor added components whose addresses have been stored in memory.

25 Perlman et al., U.S. Patent No. 5,309,437 relates to transmitting data between segments of a common LAN, whereby a host processor in any segment can address a

BEST AVAILABLE COPY

2

processor coupled in another segment of the LAN. In particular, Perlman et al. disclose a bridge-like internet protocol router which functions as a bridge but at the network layer level of addressing. The claims in
5 Perlman require "bridge-like means --- uses network addresses and network segment addresses" to forward messages between networks or network segments.

Nickolls et al., U.S. Patent No. 5,243,699 generally discloses a scaleable system and, in particular, an
10 input/output system in the form of a multi-stage router interconnection network for interconnecting an array of processor elements (computers). Address information is stored in a register and is used for establishing the routing path through the input/output system from source
15 router ports coupled to the processors, to a plurality of target router ports. This system is scaleable in the sense of selecting a set of processors in parallel and of operating the input/output system in scaled fashion to realize a cost efficient operation. The claims in
20 Nickolls are variously directed to detailed features of such a parallel processor system, such as a router which has a plurality of router ports coupled to the processors and a plurality of target router ports.

Sidhu et al., U.S. Patent No. 5,150,464 also relates to a
25 method of assigning addresses to devices coupled to a

BEST AVAILABLE COPY

3

LAN. Sidhu assigns a network address and a number identifying a particular member of the network. Each of the claims of this patent call for assigning address numbers to a plurality of subsets of devices connected to the network by selecting a value for each of one or more subsets and then checking whether that value had been previously assigned to an entity or device within its subset.

Fitzmeyer, et al., U.S. Patent No. 4,749,992 discloses a remote utility reading and control system which includes a central utility use data bank which communicates by communications link with a plurality of relay modules located at power sub-distribution transformers.

Romagosa, U.S. Patent No. 4,453,213 discloses complex methods of checking for errors in data transmission, involving a network of modules between which data is transmitted. The module which detects the data transmission error terminates the operation of the transmitting module and alerts an "arbitration" module," which in turn reroutes the erroneous data to a maintenance processor. The maintenance processor analyses the error and identifies the module transmitting the erroneous data and the module addressed to receive it. The claims in Romagosa are limited to means responsive to the detection of a data transmission error

BEST AVAILABLE COPY

4

for rerouting the transmitted data away from the originally addressed module.

Notwithstanding the aforementioned references, Applicant is not aware of any reference which teaches or speaks to the totality of invention disclosed and claimed in the present application.

SUMMARY OF THE INVENTION

It is therefore an object of the present invention to provide for a scaleable or adaptive communications device for serving as a gateway between a local area network (LAN) and a wide area network (WAN).

It is a further object of the present invention to provide for a bidirectional communications device for communication between a LAN and a WAN.

It is a still further object of the present invention to provide for a communications device between a local area network which takes the form of a segment of a power line carrier (PLC) and a WAN.

Lastly, it is an object of the present invention to provide for a communications device for serving as a gateway between a LAN and WAN which is capable of being enhanced to expand the gateway to undertake higher level functions, including computing, analyzing and transmitting collected data

BEST AVAILABLE COPY

5

to achieve a variety of preprogrammed objectives.

These and other objects of the invention will become apparent to one skilled in the art from the following more detailed disclosure of the invention.

5

In accordance with these and other objects, the present invention is directed to a scaleable or adaptive communications device or gateway for serving as a bridge between a local area network (LAN) and a wide area network (WAN). In one preferred embodiment of the invention the LAN takes the form of a segment of a power line carrier (PLC) as defined by transformers on either end.

The system architecture of the device of the present invention permits the device to act as a gateway or interface between a LAN (as coupled to LAN Controller) and a WAN (as coupled to the TTL to RS232 Converter). This architecture permits any number of elements, such as for example, a high powered computer, a video controller etc., to be connected via an expansion bus. The benefit of such a feature is to "scale" or increase the functional capabilities from a modest level as a gateway to a very sophisticated controller as may be required. In the preferred embodiment of the present invention, such a gateway would be connected to a secondary section of a power grid and communicate over that section with

BEST AVAILABLE COPY

6

meters for measuring such electrical parameters as energy, current and voltage at various grid points. Of course, this gateway may be used in a variety of other data gathering systems and communication could, for example, be effected by

5 RF transceivers.

In a further aspect of this invention, an expandable interface for coordinating the transmission of data between first and second network system. The interface comprises a

10 coupler to connect selected of a plurality of expansion devices, and at least one expansion device adapted to be connected to the coupler. The one expansion device includes a first programmable processor. The expandable interface includes a second programmable processor, which comprises at

15 least first, second and third ports connected respectively to the first network system, the second network system and the coupler. The second programmable processor is multitasked programmed to perform at least first, second and third tasks. The first task is to receive data at any one of the first,

20 second and third ports. The second task is to select another one of the first, second and third ports. The third task is to route data to the selected other port.

In a further aspect of this invention, the second

25 processor includes first, second and third ports, which comprise respectively first, second and third interrupts, and

BEST AVAILABLE COPY

7

first, second and third buffers connected respectively with the first, second and third interrupts. The second processor is further multitasked programmed to perform a fourth task. The fourth task is to detect the presence of data at any one
5 of the first, second and third tasks and to transfer a unit of said data appearing at the one port to its corresponding buffer.

In a still further aspect of this invention, a system is
10 disclosed for collecting data from a plurality of groups of data transponders. Each transponder is disposed at a site for measuring a value of a parameter at its site. The system comprises at least one first interface, and a plurality of second interfaces, at least one first data transmitting medium
15 and at least one second data transmitting medium. The second data transmitting medium is coupled to a plurality of second interfaces. A plurality of third data transmitting media is provided, each of the third data transmitting media is adapted to be coupled to a corresponding group of transponders. The
20 system further includes a system data base coupled to the first data transmitting media that receives and stores therein
~~values of parameters from the sites. The first interface is~~
coupled to the first data transmitting medium and to the second data transmitting media. The first interface includes
25 a first processor programmed to perform at least a first task. The first task provides a plurality of first interrogation signals via the second data transmitting medium to a

BEST AVAILABLE COPY

8

corresponding one of said plurality of second transponders.

The third data transmitting medium is adapted to be coupled to the plurality of transponders. Each of the second interfaces of their plurality is coupled to the data transmitting medium

5 and to a corresponding one of the plurality of third data transmitting media. Each second interface includes a second processor programmed to perform at least a second task; each second task provides a group of interrogation signals via a corresponding one of the plurality of the third data

10 transmitting media to respective ones of a group of transponders to actuate its transponder to measure the value of its parameters.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a high-level functional block diagram illustrating how a gateway or scaleable control device (SCD) in accordance with the teaching of this invention may be included within an interface, which in turn is incorporated within a bidirectional data communications system;

FIG. 2 is a more detailed functional block diagram, which shows how the interface and its gateway as shown in FIG. 1 are included within an illustrative embodiment of the communication system, which facilitates the bidirectional flow of data between a selected one of plurality transponders as incorporated into a local area network (LAN) and a central controller included within a wide area network (WAN) and is adapted for power control and monitoring;

FIG. 3 shows a particular embodiment of the interface as first shown in FIG. 1, which has been adapted for use with the system shown in FIG. 2;

FIG. 4 is a detailed functional block diagram of the scaleable communications device as generally shown in FIGS. 1-3;

FIG. 5 is a more detailed functional block diagram of a passive bus, as generally shown in FIG. 4, for bidirectionally transmitting data there along and including a plurality of

10

slots connected to the bus for selectively receiving a plurality of expansion devices;

FIG. 6 is a functional block diagram illustrating the function and relation between that computer program

5 (illustrated by blocks) executed by a microcontroller unit (MCU) and the controller, illustrated by circles, for interfacing with the LAN, the WAN, as shown in FIGS. 1-4, and the passive bus, as shown in FIGS. 4 and 5;

FIGS. 7A, B and C are respectively a more detailed flow
10 diagram of a task scheduler software module, as generally shown in FIG. 5, for detecting the presence of a data packet and routing that packet to the correct destination, a more detailed flow diagram of the process for routing an incoming data packet based variously upon that address borne by the
15 packet and the source of that packet, and a flow diagram illustrating in detail the steps performed by a priorities software module to set the priorities by which the various operations of the SCD are carried out;

20 FIGS. 8A and B are respectively a flow diagram of a scaleable expansion protocol module program stored in a system memory and executed by the MCU, generally shown in FIG. 4, for determining the presence or absence of the intelligent expansion device which may be connected to the passive bus, as
25 shown in FIGS. 4 and 5, and a functional block diagram of the

BEST AVAILABLE COPY

11
manner in which a plurality of data controllers, each
associated with a corresponding expansion device, are
interconnected with each other to permit data to be
transmitted from a selected one of the expansion devices to
5 another;

FIG. 9 is a diagram of the structure of a data packet to
be transmitted bidirectionally between the WAN, as illustrated
in FIGS. 2 and 4, and the SCD, as shown in FIGS. 1, 3 and 4;

10
FIG. 10 is a flow diagram of the program module executed
by the central controller for detecting and recovering (or
correcting) errors in data transmitted between the LAN of
FIGS. 2 and 4, and the SCD of FIGS. 1 and 4; and

15
FIG. 11 is a functional block diagram illustrating both a
first illustrative embodiment of this invention in the form of
a two tiered data transmission system and a second illustrative
embodiment of this invention in the form of a three tier data
20 transmission system.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring now to the drawings and in particular to FIG.
1, there is shown a bidirectional data communication system
25 identified by the numeral 10. The system 10 comprises a relay
or interface 20, which is interconnected by a first

12

communication medium 19 to a central controller 18 and by a second communication medium or network 16 to a remote element or transponder 22. The first and second communication media 19 and 16 may be selected from any of a variety of media including RF or wireless, data cables, cable TV, microwave, telephone, power line carriers (PLC) etc., as long as the first medium 19 is different than the second medium 16. In the following illustrative example, the first medium will take the form of an RF link 19 and the second medium a PLC 16. In this sense, the data communication system 10 is a hybrid, in that it employs two different media. Appreciating that data transmitted over the first and second media 19 and 16 may be transmitted in different data formats and rates, the interface 20 is coupled therebetween to accommodate the different formats and/or rates, whereby a bidirectional data path is opened transparently between the remote terminal 22 and the central controller 18.

FIG. 1 further illustrates the relay 20 as including a gateway or a scaleable control device or gateway 31, a first medium or power line controller 34 for interconnecting the SCD 31 to the central controller 18, and a second medium controller or transceiver 30 for interconnecting the SCD 31 to the remote terminal 22. The first and second medium controllers 34 and 30 adapts or modulates the data to be transmitted over its respective first and second media 19 and

BEST AVAILABLE COPY

13

16. As will be explained in detail below, the SCD 31 decodes or converts the incoming signals from that data format or frequency to a common, predetermined format and rate. In accordance with the teachings of this invention, the SCD 31 performs numerous other functions and is adapted by the selective inclusion of expansion devices to perform any number of other services for the bidirectional data communication path and/or relate devices as may be coupled to the SCD 31.

As shown in FIG. 2, the SCD 31 is adapted for use in the bidirectional data communication system 10 and, in a particular embodiment, a power control and monitoring system 10, which is illustrated in the enclosed functional block diagram marked FIG. 2. As shown in FIG. 3, the SCD or gateway 31 is an element of a relay or interface 20, which in turn is an element of the system 10. The system 10 permits the bidirectional transmission of data between a central controller 18 and each of a plurality of transponders 22. The communication system between the central controller 18 and the transponders 22 includes in one illustrative embodiment of the invention a first data transmission medium in the form of a wireless or RF link 19 system and a second data transmission medium 16 in the form of a power line carrier. In an illustrative embodiment of this invention, this RF system may take the form of Motorola's iDEN 900 MHz digital trunking radio system. A part of that Motorola system includes the

BEST AVAILABLE COPY

14

transceiver 34, which is included within each of a plurality of the relays (or interfaces) 20-1 to 20-n. It is understood that the Motorola system further includes a similar transceiver included in the central controller 18, whereby the first transmission medium 19 is established between the central controller 18 and each of the interfaces or relays 20-1 to 20-n. The first transmission medium 19 is also referred to as an RF link 19.

Each of the relays 20-1 to 20-n is further connected to a corresponding one of a plurality of secondary transmission lines 16-1 to 16-n. The second transmission medium 16 is also known as Power Line Carriers (PLC) or power lines. In turn, each PLC or power line 16-1 to 16-n is connected to a corresponding subset of transponders 22-1 to 22-n and via a corresponding distribution transformer 14 to a main power line 12. Thus each of the power lines 16 is connected to the secondary winding of the transformer 14 and may be referred to as a secondary line 16. As shown in FIG. 2, the central controller 18 communicates with each of the plurality of the relays 20-1 to 20-n and its corresponding one of the secondary lines 16-1 to 16-n. The transponders 22 may take the form of an automatic meter reader to be inserted within a residence or factory to measure an utility or of a device for controlling the removal and application of that utility to a consuming device. As shown in dotted line in FIG. 2, the RF link 19 and

BEST AVAILABLE COPY

15

its related transceiver 34 comprise a wide area network (WAN) 28. On the other hand, the secondary lines 16 and the transponders 22 connected thereto comprise a local area network (LAN) 30.

5 As shown in FIG. 3, the relay or interface 20 is particularly adapted for use in a power control and monitoring system, which includes the transceiver 34, a gateway or SCD 31 and a CEBus power line controller 32, which in turn is coupled to its secondary line 16. Basically, the power line
10 controller 32 operates like the transceiver 34 to receive from and apply data signals to the bidirectional path. In particular, the transceiver 34 operates with the second medium 19, i.e., the RF link between each of the relays 20 and the central controller 18, to receive from and transmit data to
15 the RF link. In like fashion, the power line controller 32 is connected to its secondary or power line 16 to receive and to transmit data over the line 16 from and to an addressed one of the transponders 22. The SCD 31 is interconnected between the transceiver 34 and the power line controller 32, and functions
20 basically to permit the transmission of data in both directions, i.e., first toward an addressed transponder 22 and then toward the central controller 18, as will be explained in detail below. In one embodiment of this invention, the SCD 31 merely acts as a conduit for this bidirectional flow of data
25 without any processing and/or storing of the transmitted

BEST AVAILABLE COPY

information or data.

The manner in which the power line controller 32 transmits data onto and from the secondary line 16 is set in one illustrative embodiment in accordance with the standards published by the Electronics Industry Association (EIA). These standards are known as the CEBus Standard EIA-600. A significant requirement of these standards is that the carrier frequency injected on the secondary line 16 by the power line controller 32 varies in the range of 100KHZ to 400KHZ. It will be appreciated that the signals imposed on the secondary line 16 may be set or modulated in accordance with other standards and technologies such as X-10, LonWorks™, or other proprietary protocols.

FIG. 2 illustrates n secondary lines 16-1 to 16-n, each a separate network to which a corresponding subset of the transponders 22a-22n is connected. Data is transmitted from the central controller 18 to a selected one of the relays 20 and, therethrough, to a selected one of the transponders 22 which is connected to its corresponding secondary power line 16. For example, data directed to the transponder 22-2n would be first transmitted to the relay 20-2 and second to the transponder 22-n. Depending on the type of transponder 22, a utility measurement is taken. Alternatively, a control switch could be thrown to apply or remove power, and the status of the switch, on or off, determined. Then, a message is sent

BEST AVAILABLE COPY

17

from the transponder 22-2n bearing either the utility usage data or the switch status via the related relay 20-2 to the central controller 18. Thus, bidirectional transmission of data is established between the central controller 18 and a
5 selected, addressed transponder 22.

There are at least two problems posed by the power monitoring and controlling system 10. First, cross-talk or interference may occur between closely related secondary power lines 16. Without a suitable isolating or blocking element,
10 the data transmitted on one secondary line 16-1 may be transmitted across its distribution transformer 14-1, along a section of the main power line 12 and across another distribution transformer 14-2 to its secondary line 16-2. Thus, undesired signals, i.e., cross-talk, are imposed on the
15 secondary line 16-1 and most probably will interfere with data to be transmitted over line 16-2. Unless the closely related secondary lines 16-1 and 16-2 can be isolated, it would be necessary to employ a relatively large number of addresses, one for each of all of the transponders or peripheral devices
20 22. On the other hand if you can isolate each secondary line 16 from the main power line 12 so that data can not then pass between any of the lines 16 and the main power line 12, then the number of addresses can be limited to the maximum number of transponders 22 which may be connected to any one of the
25 secondary lines 16-1 to 16-n. As a result, addresses may be

BEST AVAILABLE COPY

18

reused for each group of transponders 22 associated with a particular power line 16 and its transformer 14. The addressing mechanism as would be included within the SCD 31 and the central controller 18 may be simplified and its cost
5 reduced.

Data isolation between the secondary lines 16 is achieved when a relatively high frequency carrier signal is imposed by the CEBus power line controller 32 on the lines 16. This high frequency carrier signal is thus applied to the secondary side
10 of its distribution transformer 14, whereby its impedance is substantially increased to substantially block the transmission of data across the transformer 14 between the secondary and primary sides thereof. The power line controllers 32 have adopted the CEBus PLC standard, which
15 specifies that the carrier signal coupled to the secondary line 16 be of a high frequency in the range of 100 KHz to 400 KHz. At such frequencies, the resulting impedance across the distribution transformers 14 is sufficiently high to block data transmission across the transformers 14 to the main power
20 line 12. Thus, the resultant isolation blocks the transmission of data from one line 16 to another to prevent cross-talk and to limit the required number of addresses assigned by the central controller 18 to the maximum number of transponders 22 which maybe coupled to a secondary line 16.
25 Of course, each transponder 22 also requires an address for

BEST AVAILABLE COPY

19

its relay 20.

The accessing of a particular transponder 22 of the system 10 begins at the central site or controller 18. In one illustrative embodiment of this invention, transponder
5 interrogation of the system 10 is initiated by its central controller 18, which first selects a particular address and other data routing information for the transponder 22 to be interrogated and, then, extracts from a directory maintained at the controller 18 the address of the selected transponder
10 22. The transponder address is a composite address which comprises first and second parts. The first part is the address of that relay 20 which is coupled to the same secondary line 16 as the selected transponder 22. The second part is the CEBus address of the selected transponder 22. In
15 the system, each transponder 22 on a particular secondary line is assigned a unique address. In some versions of the system, like sets of addresses may be given to corresponding subsets of the transponders 22 as long the address of each transponder 22 connected to a given line 16 is unique.

20 Next, a first signal bearing the first and second address parts is transmitted over the radio link 19 towards that relay 20, which corresponds to the first address part. Next, the addressed relay 20 merely couples the radio link 19 to that relay's secondary line 16 whereby a communication path is
25 opened between the central controller 18 and the selected

BEST AVAILABLE COPY

30

transponder 22. Once the communication path is opened to the addressed transponder 22, that transponder 22 may perform a "hand-shake" with the central controller 18 to inform the controller 18 that the communication path to the selected transponder 22 has been opened. Next, the central controller 18 sends over the opened communication path to the selected transponder 22 a second or control message, which is encoded only with the second address part of the selected transponder address, i.e., that address particularly identifying the transponder 22. The addressed transponder 22 responds to the received control message whereby a utility meter reading is taken or a control switch thrown. In the final step, the selected transponder 22 completes the communication process by transmitting a signal over the still open communication path to the central controller 18, confirming that the control order, i.e., taking the utility measurement or throwing a switch, had been carried out. Upon receipt of the confirming message, the central controller 18 closes the opened communication path.

20 In the illustrative embodiment of the the communication protocol described above, the relay 20 plays a minor role. It merely opens the communication path between the radio link and its secondary line 16. The relay 20 neither initiates the transmission of any signal to the selected transponder 22 or
25 to the central controller 18, nor processes and/or stores any

BEST AVAILABLE COPY

31
data or information transmitted over the bidirectional path.

With regard now to a detailed functional block diagram of FIG. 4, there is shown the details of the scaleable communications device (SCD) or gateway 31 according to a preferred embodiment of this invention. The SCD 31 functions as an interface to control the bidirectional data flow between at least two different data transmitting systems, e.g., the wide area network (WAN) 28 and the local area network (LAN) 30. The format and/or carrier signal to be modulated by the data, typically digital, appearing on the LAN 30 and the WAN 28 may be different in terms of format and rate (or frequency) of transmission.

Still referring to FIG. 4, the SCD 31 includes a microcontroller unit (MCU) 112 for executing various software, as will be explained, to facilitate the bidirectional flow of data between the WAN 28 and the LAN 30, and renders the SCD 31 adaptable or scaleable to variously include added components beyond that basic configuration shown in FIG. 4, whereby the SCD 31 is capable of implementing many different data processing operations at varying levels of complexity. Such scaleability is imparted to the SCD 31 by a passive bus 114, which is coupled to the MCU 112 by a double buffered data latch 124 which functions generally to buffer data transmitted between the bus 114 and the MCU 112 and to permit the bus 114 and MCU 112 to transmit and process data independently of the

BEST AVAILABLE COPY

22

other element. In FIG. 4, there is shown at least 3 data paths or networks: 1) the first transmission medium of the WAN 28 in the illustrative form of the secondary power lines; 2) the second transmission medium 16 of the LAN 30 in the illustrative form of the RF link; and 3) the passive bus 114.

Each path 16, 19 and 144 is appropriately interfaced to a port or input of the MCU 112. In an illustrative embodiment of this invention, these ports take the form of the interrupts of the MCU 112 which are identified by the numerals 133, 117a and 137. These ports or interrupts 133, 117a and 137 are respectively coupled via the WAN controller 32 to RF link 19, via the double buffered latch 124 to the passive bus 114 and via the LAN controller 32 to the power line 16. Each interrupt 133, 117a and 137 has a corresponding queue 113a, b and c respectively for temporarily storing data inputted via their respective interrupts. As shown in FIG. 4, it is contemplated in one embodiment of this invention that queues 113 may be included within the structure of the system memory 138. Further as the name implies, the double buffered latch 124 includes at least first and second buffers 124 a and b, each interconnected between the passive bus 114 and the MCU 112. An address decoding circuit 126 is connected between the passive bus 114 and the MCU 112 to decode the input/output addresses necessary for communication therebetween. Data lines 117a and 117b respectively permit a data flow between the passive bus 114 and the data latch 124, and between the data latch 124 and the MCU 112. Read/write control lines 119a

BEST AVAILABLE COPY

13

and 119b interconnect respectively the data latch 124 to the MCU 112 and to the address decoding circuit 126. Interrupt lines 115a and 115b respectively interconnect the MCU 112 to the passive bus 114 and to the address decoding circuit 126.

5 Address lines 121 interconnect the passive bus 114 to the address decoding circuit 126.

The structure and operation of the passive bus 114 are more fully illustrated in FIG. 5. The bus 114 includes data
10 lines 118a and b, connected in parallel to a plurality of slots 120-1 to 120-n, for transmitting data respectively to and from each of the slots n to the MCU 112 via the data latch 124. To augment the functions of which the SCD 31 is capable, one or more of a plurality of expansion devices 122-1 to 122-n
15 is inserted into any one of the slots 120-1 to 120-n. One and possibly more of the expansion devices 122 must be a smart or intelligent device. In the context of this invention, smart or intelligent means that the device 122 must be capable of controlling the data transmission along the passive bus 114.

20 In particular, each intelligent expansion device 122-1 has its own programmable processor (not shown). Typically the processor operates asynchronously with respect to the MCU 112.

The processor of each of an intelligent expansion device 122-1 operates asynchronously with respect to the MCU 112 and,
25 like the MCU 112, is programmed to perform the hardware protocol described below to operate the latch 124 to transfer

BEST AVAILABLE COPY

24

data between the processor of the intelligent expansion device 122-1 and the MCU 112 in an asynchronous manner. The bus 114 is also connected, as shown in FIG. 5, to the buffered latch 124 which permits data to be transferred one byte at a time between the bus 114 and the MCU 112. The intelligent expansion device 122-1 is distinct from the MCU 112 because the data flows along the passive bus 114 and through the SCD 31 are not synchronous.

As shown in FIG. 5, each of the passive expansion devices 122-2 to n and the intelligent expansion device 122-1 includes, in one illustrative embodiment of this invention, a data controller 123. Generally, the data controllers 123-1 to n control the transmission of data between each of the expansion devices 122-1 to n and the passive bus 114. In particular, only one of the expansion devices 122-1 to n may transmit or receive a signal along the passive bus 114 at a time. The data controllers 123-1 to n control the flow of data over the data lines 118a and b and keeps track of the other expansion devices 122 by remembering their preassigned addresses. There is shown in FIG. 8B a functional block diagram of how the data controllers 123-1 to n, which includes a computer in the illustrative form of a microprocessor, are interconnected to permit only one of the expansion devices 122-1 to n at a time to transmit or receive signals via the passive bus 114.

Though not shown in FIG. 5, the intelligent device 122-1 includes a controlled processing unit (CPU), a memory,

BEST AVAILABLE COPY

25

additional hardware and "device driver" software executed by the CPU to allow the intelligent device 122-1 to transmit data over the passive bus 114 to the MCU 112. In the context of this invention, passive means at least lacking capability of
5 controlling the data transmission of data along the bus 114.

In an illustrative embodiment of this invention, the passive bus 114 is a 16-bit, ISA type bus, and the slots 120 are 98 pin 16-bit ISA compatible slots. The passive bus 114 allows the intelligent ISA expansion device 122-1, such as a
10 single card ISA computer, to be added to the bus 114 at any time. The single card computer can communicate with the bus 114 and share resources with it. A simple protocol allows communications between the MCU 112 and the single card computer, and allows each to recognize the other's existence.
15 Communications are established, as shown in FIG. 4, via interrupt lines 115a and 115b between the MCU 112 and the passive bus 114 on which the intelligent expansion device 122-1 resides. Both the ISA bus 114 and the MCU 112 are respectively connected by interrupt lines 115a and 115b, which
20 enable bi-directional communications therebetween without complex protocols. The bi-directional communications are accomplished via pre-determined input/output (I/O) addresses.

The MCU 112 is set to respond to a particular I/O address and the intelligent device 122-1 that can actively share the ISA
25 bus 114 and communicate with the MCU 112 at this address.

BEST AVAILABLE COPY

26

Since the bus 114 contains more than one passive ISA slot 120, the passive expansion devices 122-2 to n may take the illustrative form of an ISA modem, ISA Video Card, ISA disk drive controller or any other ISA device. This allows maximum scaleability of the SCD 31 since the SCD 31 can be deployed without any ISA expansion devices 122 as a simple communications gateway or it can be deployed with several ISA devices to form a network controller or remote terminal unit. As shown in FIG. 5, one of the slots 120-1 to 120-n is adapted to receive a passive expansion device 122-n, which takes the form of an input/output (I/O) board. The I/O board 122-n permits the input of data from a variety of sources, particularly those in the vicinity, as well as to output signals.

15 The address decoding circuit 126, in such an illustrative embodiment as shown in FIG. 4, decodes the ISA I/O addresses necessary for communication between the ISA bus 114 and the MCU 112. The SCD 31 can be modified to respond to any valid ISA I/O address. Currently the address is selectable via shorting blocks with available addresses being 200\$, 208\$, 210\$, 218\$, 220\$, 228\$, 230\$ and 238\$. In addition to enabling the SCD 31 to decode, the address decoding circuit 126 also provides some of the signals necessary to operate the double buffered latch 124. An I/O write from an ISA expansion device 122 to a valid SCD address allows data to be written

BEST AVAILABLE COPY

27

from the ISA data bus 114 to the data latch 124 and also initiates a hardware interrupt to the MCU 112 signalling the MCU 112 to fetch the data awaiting in the latch 124.

Likewise, a I/O read from one of the ISA expansion devices 122 to a valid SCD address allows data to be written to the ISA data bus 114 from the data latch 124 by the MCU 112. This type of buffering arrangement is necessary to allow the SCD 31 to accomplish timing critical tasks while interfacing with the ISA data bus 114.

10 In this illustrative embodiment as shown in FIG. 4, the data latch 124 is an 8-bit double buffered latch, serving as a depository of data that is communicated between the MCU 112 and the ISA data bus 114. The bus 114 is capable of transmitting data bi-directionally with each direction being
15 able to latch a single byte of data. The latch 124 allows ISA expansion devices 122 and the MCU 112 to deposit data and continue task processing without waiting on the expansion device 122 to immediately acknowledge receipt of data via hardware. The data latch 124 illustratively comprises four 8-
20 bit latches/buffers (not shown in the drawings) with two latches/buffers being utilized for data flowing from the ISA passive bus 114 to the MCU 112 and two for the data moving from the MCU 112 to the ISA data bus 114. As will be explained, the latch 124 is individually controlled either by
25 the MCU 112 or the address decode circuit 126.

BEST AVAILABLE COPY

28

As shown in FIG. 4, the 8-bit micro-controller 112 is the heart of the SCD 31. The MCU 12 controls the entire device. Also present in the MCU 112 are peripheral and other necessary interface devices. Some of these include, but are not limited

5 to:

Multiple bi-directional input/output (I/O) registers
External memory address lines
Internal RAM, ROM and EEPROM
Asynchronous Serial Communications Interface
10 Analog to Digital (A/D) Converter
Maskable and non-maskable hardware interrupts
Multiple internal counter registers necessary for
timing critical tasks

A system memory 138 includes a Static Random Access
15 Memory (SRAM), an Electrically Erasable Programmable Read Only
Memory (EEPROM) and an Erasable Programmable Read Only Memory
(EPROM) or Read Only Memory (ROM). Part of the SRAM may be
battery backed with a ten year battery and includes a real
time clock/calendar in its upper address range. All of the
20 operating system is in this illustrative embodiment placed in
the EPROM or ROM. The EEPROM is used for variables that
rarely change and are nonvolatile and critical to the proper
functioning of the device. Examples of this type of variable
is a device address which is assigned when the unit is first
25 installed in the environment in which it is to operate such as

BEST AVAILABLE COPY

29

the local area network (LAN) 30.

In addition to being able to communicate with the SCD 31 via the ISA bus 114 and/or introduce an element by an 8-bit expansion bus 114 which will be discussed later, the MCU 112 also has incorporated a standard RS-232C asynchronous serial communications port or interrupt 133, which is coupled to a RF or WAN controller 34 to establish a bidirectional flow of data to and from the WAN 28 via the controller 34. The port 133 is formed by using a TTL level serial port on the MCU 112 along with a control line to provide flow control of data on a communicating medium, e.g., the radio link 19 as shown in FIG. 2. The TTL level signals are transformed by the controller 34 to RS-232C level signals via a RS-232C/TTL driver chip. Flow control is accomplished via the "clear to send" (CTS) and "ready to send" (RTS) pins. While this port 133 can be used to communicate with other external devices such as lap top computers, the port 133 is most often used as the connection to the controller 34 for the wide area network 28.

Since the SCD's MCU 112 has multiple analog to digital (A/D) converter inputs, the SCD 31 has been employed to monitor certain voltage levels. Among these voltages are the individual phases of the utility secondary service on which the SCD 31 is placed. Since the raw 90 - 310 VAC phase voltages to be sensed and processed by the SCD 31 cannot be directly connected to the SCD's MCU A/D port, a conditioning

BEST AVAILABLE COPY

30

analog to digital (A/D) circuit 140 is utilized to supply the MCU A/D port with voltages proportional to the actual AC line voltage. The circuit 140 includes a resistive divider network, rectifier, filter capacitor and discharging resistor.

5 The conditioning circuit 140 is coupled by lines 135 to an A/D port of the MCU 112. The conditioning circuit 140 is used to monitor, for example, the following voltage levels:

Phase A to Neutral AC input voltage

Phase B to Neutral AC input voltage

10 Phase C to Neutral AC input voltages

Unregulated DC power supply bus

13.7VDC power supply bus

12VDC power supply bus

5VDC power supply bus

15 13.7 WAN transceiver power supply bus

In order to correctly produce a digital representation of the voltage sensed by the conditioning circuit 140, the MCU 112 requires two reference voltages to which that voltage of the digital signals appearing on the A/D lines 135 are compared.

20 The two references are comprised of an upper and lower voltages which are supplied by precision, voltage references possibly buffered by OP-amps in order that it be able to drive the highly capacitive inputs to the A/D voltage reference lines. The references are also temperature stabilized to help

BEST AVAILABLE COPY

31

offset error introduced by temperature variances.

An 8-bit MCU data bus is coupled to an expansion bus 141 of the SCD 31. This expansion bus 141 is a 14 pin header connector which contains the 8-bit data bus, three I/O control lines as well as +5VDC and digital ground. Using this bus 141, the SCD 31 can be interfaced directly to auxiliary circuits in order to expand the functionality of the SCD 31. For example, an I/O board 143 may be coupled to the expansion bus 141 to permit the input of signals and the output of signals, e.g., the output of a signal to an alarm device.

The SCD 31 is designed, as shown in the embodiment illustrated above with respect to FIGS. 2 and 3, to operate with either or both of the Radio Frequency (RF) link 19 and the Power Line Carrier (PLC) line 16. To that end, the SCD 31 is interfaced with the consumer electronics bus (CEBus) RF and/or PLC controllers 34 and 32. These controllers perform the bottom two layers (physical and data link layers) of OSI's standard open protocol stack. These controllers 34 and 32, as shown in FIG. 4, are attached to the MCU's 8-bit data bus as well as other control lines which allows the MCU 112 and the controllers 34 and 32 to perform two way data flow control via a hardware "handshaking" routine.

The LAN controller 32 is initialized for operation by the MCU 112. After initialization, the LAN or power line

BEST AVAILABLE COPY

32

controller 32 sends and receives data from the SCD 31 as allowed in the initialization. There are two standard modes of operation in which the LAN controller 32 can operate:

Standard and Monitor. In its standard mode, the LAN

5 controller 32 passes data to the SCD MCU 112 only if that data is addressed specifically to the SCD 31. Each SCD 31 is assigned a LAN address prior to installation. In its monitor mode, all LAN data is routed to the SCD's MCU 112 and the MCU 112 must sort out the desired data.

10 When the SCD 31 is configured to communicate to the PLC or secondary line 16 and its local area network (LAN) 30, a power line coupling circuit 146 is employed to allow the signals produced by the LAN controller 32 to be superimposed upon the AC power line 16. The coupling circuit 146 is
15 arranged such that the PLC signal is coupled onto all AC supply phases that might be present. The coupling circuit 146 includes a coupling transformer, coupling capacitors and protective diodes (none shown in drawings). The SCD 31 is connected to the secondary line 16 using the coupling
20 transformer and coupling capacitor. The protective diodes are
5 watt, 4.3V Zener diodes placed back to back across the secondary of the coupling transformer.

A data latch in addition to the 8-bit double buffered data latch 124 is also present on the SCD 31. The function of

BEST AVAILABLE COPY

33

this latch (not shown) is to provide a means for the SCD 31 to write data to a set of eight LEDs 148, which indicate the status of certain SCD functions as well as allow indication of data flow on the wide area network (WAN) 28 as well as the local area network (LAN) 30. A "power ON/MCU status" LED provides a visual indication that the SCD 31 is operating by turning this LED on and off about once every second, for example. An "Active ISA device installed" LED provides a visual indication that the passive bus 114 is operational by turning this LED on and off about every 5-10 seconds depending on the hand-shake packet from the intelligent device 112-1. An "Active ISA device executing task scheduler" LED is a visual indication that an intelligent device 122-1 (as described above) is sending data to the SCD 31 or is receiving data from the SCD 31. A "LAN data received" LED is a visual indication that CEBus packets are being received by the SCD 31 on the power line 16. A "LAN data transmitted" LED is a visual indication that the SCD 31 is transmitting CEBus packets on the secondary power line 16. A "SCD linked with WAN device" LED is a visual indication that the SCD 31 opened or established the bidirectional data communication path to the central controller 18. A "WAN data received" LED is a visual indication that the SCD 31 is receiving data packets over the RF link 19. A "WAN data transmitted" LED is a visual indication that the SCD 31 is transmitting data packets to the

BEST AVAILABLE COPY

34

WAN 34.

Figures 2 and 3 illustrate the standard SCD or gateway 31 and the relays 20-1, 20-2 --- 20-n, which periodically access selected of the peripheral devices or transponders 22 coupled to its power line 16 to collect data therefrom and to upload the collected data to its SCD 31. Referring to Figure 4, the MCU 112 executes a data collection program as stored in the memory 138 to access data collection over the power line 16. If large quantities of data were to be collected and/or data was to be collected from a large number of peripheral devices 22, a computer or smart controller of greater capacity than the MCU 112 may be incorporated into the SCD 31, by connecting an intelligent expansion device 122-1, which contains a relative high power computer, to the passive bus 114 as shown in Figure 5 to thereby significantly increase the computing power of the SCD 31. Additional memory to store or concentrate the collected data may be included in the expansion device 122-1 or a second expansion device 122-2.

20 Compared to the three tiered architecture of the data transmission system shown in Figure 11, that of Figure 2 is simple. Figure 2 shows a first illustrative embodiment of this invention, i.e., a simple network architecture comprised of the RF link 19 interconnecting a single central controller or front end computer 13 with a plurality of standard relays 25

BEST AVAILABLE COPY

35

20-1, 20-2 --- 20-n, a like plurality of power lines 16 and a plurality of peripheral devices or transponders 22 connected to each power lines 16. Figure 11 shows a second illustrative embodiment in the form of the three tiered system which

5 comprises one or more "up-line" relays or "data concentrator" 20' in addition to the simple architecture of Figure 2. Each upline relay 20' is connected by a third data transmission medium 16' to at least one standard relay 20 to produce a three tiered data transmission system, i.e., the first

10 transmission medium 19, the third transmission medium 16' and the second transmission medium 16 (as compared to the 2 network system of Figure 2). Preferably, the third transmission medium 16' is coupled to a plurality of standard relay 20a-20n, each of which is coupled to a corresponding one

15 of the second transmission media 16a-16n. In turn, each of second transmission media 16a-16n is coupled to corresponding plurality of peripheral devices 22a-n. The upline relay 20' accesses each of the standard relays 20a-20n to collect and store (concentrate) data in its memory, e.g., the system

20 memory 138 of Figure 4 or a memory incorporated into an expansion device 122 of Figure 5. In other words, the upline relay 20' and its SCD are stacked above the standard relays 20a-20n and their respective SCDs.

As further shown in Figure 11, a plurality of front end

25 controllers 13a-13n are connected to the first transmission

BEST AVAILABLE COPY

36

medium 19, whereby a selected one controller 18 may take charge of the entire multi- tiered network system 10'. Only one controller 18 at a time can be in charge of the network system 10'. Even so, the controllers 18a-18n can be in communication with each other across the first transmission medium 19.

In addition, a plurality of peripheral devices 22' can also be coupled to the first transmission medium 19 of Figure 11. Appreciating that the peripheral devices 22' can be monitors or controllers, the relay 20' through its WAN controller 34 of Figure 4 can variously send a control signal over the first transmission medium 19 to one of the peripheral devices 22'1-22'n, whereby the one device 22' would apply or remove power to a particular piece of equipment.

Alternatively, devices 22'1-22'n are able to measure a parameter of the related equipment, e.g., the power or temperature parameters, and send signals back over the first transmission medium 19 indicative of that measured parameter to the relay 20'. If a measured parameter is outside its limits, an alarm signal is sent to the I/O board 122-n, the first transmission medium 19 or the second transmission medium 16 as described above.

The three tiered network system 10' of Figure 11

BEST AVAILABLE COPY

37

comprises a great number of computers or processors including all of those included in the relays 20, the central or front end controllers 18 and, in some embodiments of this invention, selected of the peripheral devices 22. In a classic sense, this network system 10' employs distributed computer processing. The front end or central controller 18 is implemented by a computer programmed with software. As shown in Figure 4, each SCD 31, which is a part of the relay 20, includes an MCU 112, which executes programs stored in the system memory 138 and, when further computer power is needed, the SCD 31 is scaled by adding a further computer in the form of the intelligent expansion device 122-1 and the passive expansion devices 122-2 to 122-n, as shown in Figure 5.

15 In an illustrative embodiment of this invention, a memory of the SCD 31 is programmed to adapt the data transmission system 10 or 10' to perform a variety of functions and services. Exception monitoring is carried out in one illustrative embodiment of the SCD 31 by either the programmed MCU 112 and/or the intelligent expansion device 122-1

described above. This monitoring involves setting upper and lower limits of a particular parameter as stored for example in the system memory 138, scanning a set of the peripheral devices 22 to measure that particular parameter and determining whether the measured parameter falls within the

25

BEST AVAILABLE COPY

38

upper and lower limits. If out side these limits, an exception is detected and an alarm is generated and sent to the customer over the first transmission medium 19, the second transmission medium 16, e.g., a power line, an RF link, the I/O board 143, or the I/O board 122-n which is connected to the SCD's expansion bus 141, as shown in FIGs 4 and 5. Illustratively, these signals are applied to turn on an alarm device, e.g., a light or sound device, at the customer's premises, whereby the out of limit condition is made known.

10

In operation, the scanned data from the peripheral devices 22 is stored in a corresponding one of the SCD's 31, until at a later time, the front end controller 18 uploads the data stored in the SCD 31 over the first transmission medium 19. For example, an operator at the front end controller 18 could initiate the uploading of a selected SCD 31. Alternatively, the front end controller 18 can initiate the uploading at a particular time of the day or month, e.g., late evenings or weekends when transmission rates may be lower.

20 Alternatively, the SCD 31 can initiate the uploading of data, when an out of limit parameter is detected and it is desired to provide an alarm message to the front end controller 18 or wherever the customer is located. In a further illustrative embodiment, uploading is initiated when a certain amount of data has been collected in the system memory 138 of the SCD

25

BEST AVAILABLE COPY

39

31. Further, the scan of the peripheral devices 22 can be made on a periodic basis, whereby the data collected and stored in the system memory 138 represents the state of a customer's equipment over a period of time. For example, it would be possible to monitor the temperature of a room or the power consumed by a machine over a period of time. The customer can also obtain that history of measurements by transmitting an upload signal from a front end controller 18 to the particular SCD 31 to obtain a history of that parameter for the past day, week or month, for example. The programming for these scanning, measuring and determining operations is distributed among a plurality of the SCD's 31, which are included within each of the relays 20a-20n or 20-1 to 20-n as shown in Figure 11, rather than a central controller 18, to reduce the number of data transmissions on the first transmission medium 19.

In one illustrative embodiment of this invention, exception monitoring may be implemented by the addition of the I/O board 122-n, which is coupled via the expansion bus 114, as shown in Figure 5. Utilizing the expansion bus 114 and the I/O board 122-n, various signals from a piece or pieces of equipment external to the SCD 31 that are desired to be monitored are interfaced to the SCD 31. Monitoring criteria are established via the SCD firmware as described above and the SCD 31 is allowed to periodically scan for alarm

BEST AVAILABLE COPY

40

conditions via the inputs from the I/O board 122-n. Should the SCD 31 find conditions residing outside of the pre-established criteria, alarms can then be issued via one of the following methods: (a) alarm signals may be provided to a customer via one of the outputs on the I/O board 122-n; (b) alarm signals may be broadcast to a customer via the first transmission medium 19 connection through the WAN controller 34; (c) alarm signals may be broadcast to a customer via the second transmission medium 16; or (d) alarm signals may be broadcast to a customer via the first transmission medium 19.

Instead of using the I/O board 122-3, the SCD 31 may communicate with one or more transponders or monitoring devices 22 residing outside the SCD 31 via the second transmission medium which illustratively takes the form of a PLC or power line 16, a radio frequency line or other suitable transmission medium. Parameters to be monitored from each remote communicating transponder or peripheral device 22 can be, but are not limited to, energy consumption (KWH), demand (KVA), reactive power (KVAR or KQ), apparent power (KVA), power factor (PF), phase voltage, phase current, power outage, power quality including voltage and current harmonics, temperature, pressure, gas and liquid flow rates, heating or cooling rates (BTU) and security alarms. Monitoring criteria are established via SCD firmware such as the system memory 133

BEST AVAILABLE COPY

41

or a memory included within one of the expansion devices 122-2b 122-n. The SCD 31 is allowed to periodically scan via the second transmission medium 16 for out of limit "alarm" conditions. Should the SCD 31 find conditions residing
5 outside of the pre-established criteria, alarms can then be transmitted via one of the first and second transmission media as explained above to alert the customer.

Further, each SCD 31 sets a priority for the execution of
10 each module or subroutine of software as shown in Figure 6. In addition, an important feature of the SCD 31 is its task scheduler 160 shown generally in Figure 6 and in detail in Figure 7A. The SCD 31 may be also be programmed to schedule and prioritize various functions and to distribute software on
15 a system wide basis as contemplated by Figure 11. As a broad principal, the front end or central controller 18 (or a selected on the plurality of controllers 18a-18n) will set the priorities and schedule the tasks for the entire system shown in Figure 11. For example, the selected front end controller
20 18 send messages to all of the gateways 20 setting the
priorities of executing their software modules and a schedule of their tasks.

Also, the SCD 31 may be programmed to perform a variety of customer services. Typically, the programming for these
25 services is stored in the system memory 136 as executed by the

BEST AVAILABLE COPY

42

MCU 112 of Figure 4, but could also be effected by the intelligent expansion device 122-1 of Figure 5. These services can include, but are not limited to, electronic clock setting after power outage, severe weather alerts, text news services, utility services pricing, i.e. real time pricing from various providers of gas, electricity and communications in a de-regulated environment, product marketing services, utility demand side management, utility load control, home banking services, home shopping, home gaming, electronic "house sitting" services, home automation services, appliance monitoring and security services.

In addition, the SCD 31 may be scaled by adding a video camera in the form the intelligent expansion device 122-1, whereby video images could be transmitted over the first transmission medium 19. There are many security applications for this embodiment.

The SCD 31 may be modified or scaled by connecting the intelligent expansion device 122-1 in the form of a server via a high speed first transmission medium 19 or second transmission medium 16, e.g., the Internet, to a customer's data terminals, illustratively in the form of a PC, and peripheral devices 22, whereby these devices in turn could be connected to the Internet. In particular, the intelligent

BEST AVAILABLE COPY

43

expansion device 122-1 could be programmed with any of the presently available Internet interface software such as Linux.

In an illustrative embodiment of this invention, the SCD 31 of the concentrator relay 20', as shown in Figure 11, could be scaled with such software, whereby the first transmission medium 19 could be implemented by the Internet, in contrast to the RF link as suggested above. A customer, whether for commercial or personal use, could in such a network architecture access the SCD 31 of the concentrator relay 20' to check the status of a variety of equipment from the kitchen stove to a manufacturing machine as would be connected to a corresponding one of the peripheral devices 22.

There is shown in FIG. 6 a computer program 148, which is executed by the MCU 112 whereby the SCD 31 operates as a gateway to control the input and output of data from and to the second transmission medium 16 via the LAN controller 32, from and to the first transmission medium 19 via the WAN controller 34, or from and to the passive bus 114 to any extension device 122 as may be inserted within the slots 120 of the passive bus 114. The computer program 148 is first divided into an input part 148a and an output part 148b and, then into a number of modules 150a and b, 152a and b and 154a and b. As shown in FIG. 6, the "a" modules perform input functions and the "b" modules perform output functions.

44

Generally, the computer modules 150, 154 and 152 respectively control the flow of data between the SCD 31, and the first transmission medium 19, the second transmission medium 16, and the passive bus 114.

5 The computer program 148 is still further divided into first and second layers based on whether certain functions of a particular element need to be treated with high priority and thus are considered to be time critical functions, or may be treated with low priority because they are not considered to
10 be time critical. Referring to the illustrative embodiment of the SCD 31 as shown in FIG. 4, certain elements like the WAN controller 34, PLC controller 32 and the double buffered latch 124 perform certain functions which are time critical and, therefore in the scheme of the program shown in FIG. 6, are
15 given high priority. High priority is needed for these elements for a number of reasons. First as noted above, the PLC or LAN controller 32 processes data which is structured according to the CEBus Power Line (PL) Standard set by the Consumer Electronic Group of the EIA; this standard is an open
20 standard protocol. According to this standard, the insertion and retrieval of CEBus data packets by the PLC or LAN controller 32 to and from the second transmission medium 16, e.g., a power line, as well as the data packet transmission between the WAN controller or transceiver 34 and the first
25 transmission medium 19, e.g., the RF link, require the

BEST AVAILABLE COPY

45

generation of precisely timed signals and, therefore, are a part of the first, time critical layer. Secondly, the PLC or LAN controller 32 is essentially a hardware component which needs and is capable of responding quickly to critically timed control signals. In this regard, the PLC controller 32 and the other prominent hardware elements, i.e., the WAN or RF controller 34 and the double buffered latch 124, operate quickly on hardware protocols and are connected, as indicated in FIG. 4, to dedicated interrupts of the MCU 112 as explained above. In this fashion, each of the elements 32, 34 and 114 is capable of acquiring immediate response from the MCU 112 by applying its interrupt.

A still further consideration of whether an element or program module is treated as time critical or not depends on the need to interface data differently, which in turn depends on how that data is formatted and in what language it is written. Data packets transmitted to and from the SCD 31 are formatted in one illustrative example according to the CEBus Standard EIA-600 and are written in a Common Application Language (CAL). In this regard, the CEBus Standard requires the use of CAL. The PLC or LAN controller 32 and RF or WAN controller 34 only transmit CAL data packets. On the other hand, communications from an external source which does not employ CAL, may comprise CAL data packets and non-CAL data packets. Thus, different interface translating is required

46

for data packets which are written in different languages, e.g., CAL and non-CAL. The software translating processes are often not time critical and, thus, can be incorporated into the second, low priority layer.

5 As shown in FIG. 6, the program modules 150 a and b operate the hardware interfaces, i.e., the PLC or LAN controller 32, to respectively control the input and output of data packets therefrom. These modules 150 a and b are apart of the first, time critical level to permit these interface
10 operations to be carried out quickly. As shown in FIG. 6, the inputting and outputting data modules 150 a and b also are used to control in a time critical fashion data transmission between the RF or WAN controller 34 and the second transmission medium 19. Like the LAN controller 32, the RF or
15 WAN controller 34 is primarily a hardware element and should be operated in a time critical fashion to ensure the rapid bidirectional data transmission between the SCD 31 and the central controller 18. The bidirectional flow of data between the remote transponders 22 and the RF or WAN controller 34
20 only includes CAL data packets and does not require any data format or language conversion. Further, it is desired not to delay the transmission of data packets between the SCD 31 and one of the remote transponders 22. The hardware protocol which throws the buffered latch 124 to permit the
25 bidirectional flow of data to and from the passive bus 114 is

BEST AVAILABLE COPY

47

also performed by high priority, time-critical software and, in particular, also by the modules 150a and b. The latch 124 and the addressing circuit 126 are essentially hardware elements and operate in a time critical fashion with the MCU 112. The fast operation of the latch 124 is important in order that the bidirectional flow of data bytes between the MCU 112 and the passive bus 114 be carried out as quickly as possible.

Still referring to FIG. 6, the buffered protocol data modules 152 a and b are written as time critical, high priority and are also apart of the first layer. Figure 6 also represents the order of signal processing that occurs in the SCD 31, beginning first in step 150a with the input of data to one of the interrupts 137, 117b and 133 from the LAN 30, the passive bus 114 and the WAN 28, respectively. As will be explained below, the processing steps occur in the shown order of steps 152a, 154a, 156, 158, 160, 162, 154b, 152b and 150b, appreciating as will be described that step 150b outputs data to one of these networks 30, 114 and 28. As will be described, this sequence of steps may be interrupted if an event of higher priority occurs, e.g., a data bit appears at the interrupt 137 from the LAN 30. The modules 152 a and b temporarily store the data packets of a message in the queues 113a, b and c, shown in FIG. 4 as a part of the system memory 153, to permit their assembly and disassembly as they are received and transmitted respectively over the second and

BEST AVAILABLE COPY

48

first transmission media 16 and 19. The assembly and disassembly must be performed quickly and the related modules 152a and b are apart the high priority, time critical software layer. Though all of the modules 150 and 152 are considered
5 high priority, the data transmitted to and from the passive bus 114 is in an illustrative embodiment processed at a faster rate than the storing and reading of data from the noted queue.

10 In contrast to the program modules 150 and 152, the modules 154 a and b operate not to initiate the operation of the RF or WAN controller 34 in its transmitter mode to transmit data packets or in its receiver mode to receive
15 packets, but rather to perform a cyclic redundancy check (CRC) on each data packet sent or received to monitor the integrity of each byte of the packet, as will be explained in greater detail below with respect to FIG. 10. In particular, the module 154 b adds a 16-bit CRC tag, whereas the module 154 a checks the CRC tag when the signal is received by the RF or
20 WAN controller 34 from the RF transceiver disposed at the central station 18. Thus, the modules 154a and b operate at a low priority to process the CRC operations.

External events, like a power outage, are detected by the voltage conditioning circuit 140 of the SCD 31. In particular
25 as shown in FIG. 4, the voltage conditioning circuit 140 is

49

coupled to the power line service conductors 142 to sense the analog voltage applied to phases A, B and C of the secondary power lines 16 comprising the second transmission medium and applies corresponding digital signals via the A/D lines 135 to the MCU 112. The power line service conductors 142 are also connected via the power line coupling circuit 146 and the PLC or LAN controller 32 to the interrupt 137 of the MCU 112, where control may be exercised in a time critical fashion. The value of the voltages as read by the circuit 140 is used to measure external events as indicated by the hardware. The recognized events are stored in an internal table used by the low-priority software. The high priority program notifies the low priority program of the external condition and the low priority program level starts a series of events to process this condition.

As shown in FIG. 6, the CAL decoder 156 processes all incoming CAL packets as they arrive via the first or second media 19 or 16. The decoder 156 interprets the packet based on a table of available functions that it can perform. If the decoder 156 is unable to interpret the packet, an error message is generated and an error packet is added to a task list 158, which is a table maintained in the system memory 138. If the message is understood and interpreted by the CAL decoder 156, it is passed directly to the task list 158. The task list 158 contains a list of packets that is handled by

BEST AVAILABLE COPY

50

the low priority, non-time critical layer of the program 148.

The packets are pulled off the list 158 in a FIFO (first-in first-out) manner and sent to a task scheduler 160, whose steps will be explained below with respect to FIG. 7A. Next, 5 the CAL encoder 162 puts all of the sub-components together from the OSI model module and creates a valid CAL packet. The task scheduler 160 routes each data packet to one of the controllers 32 or 34, or to the passive bus 114. If the packet is a response to a request originating from the LAN 30 10 or the WAN 28, the packet may be routed for example to the central controller 18 or a selected one of the remote terminals or transponders 22; otherwise the packet is passed over to the CEBus stack for further processing. The CEBus stack is a modified version of the open systems 15 interconnection (OSI) model of the International Organization for Standardization (ISO). This modified model contains only four of the original seven layers of the OSI model and provides the needed services to properly compose or extract information from a CEBus packet.

20 The task scheduler 160, shown generally in FIG. 6, is shown in more detail in FIG. 7A as a program comprised of a series of steps. The task scheduler 160 is a low-priority, non-time critical, program or subroutine, which is called by an application program executed on the MCU 112, as opposed to 25 being called by an interrupt applied to the MCU 112. In the

BEST AVAILABLE COPY

51

first step 231, a search is made of the task list 158 to determine whether any tasks have been sent to the list 158. If none, the scheduler program 160 moves to the end 240. If there is a task to be performed, and the program continues with the CAL encoder 162, as shown in FIG. 6. If there is at least one task set in the list 158, the subroutine 160 moves to step 232, where first entered task is removed in a FIFO manner to be examined in steps 234 - 246. First in step 234, the task is examined to determine whether it involves a data packet to be routed via the PLC or LAN controller 32 to be transmitted out over the second transmission medium 16. If a data packet is intended for the PLC 16, step 236 sets the route of that packet to the PLC or WAN controller 32 as shown in FIG. 6. If not a PLC or second medium packet, step 242 examines the task to determine whether it involves a data packet to be routed via the serial port and its RF or WAN controller 34 to the first transmission medium 19. If an RF packet to be transmitted over the second medium 19 is detected, step 236 routes that data packet to the WAN controller 34 to be transmitted therefrom to the central controller 18. If not an RF or first medium data packet, step 244 examines the next task in the list 158 to determine whether the data packet should be transmitted to the ISA or passive bus 114, as shown in FIGS. 4 and 5. If so, step 236 routes this data packet to the latch 124 to be transmitted one

BEST AVAILABLE COPY

52

byte at a time to the passive bus 114 by the hardware protocol which will be explained below in detail. If the data packet is not for the passive bus 114, step 246 assumes that it is intended for the MCU 112 and that data packet is so routed
5 without need to be encoded by the encoder 162. In the layered scheme of program 148 as shown in FIG. 6, data packets for such internal tasks are handled by the low-priority, non-time critical layer. Once a data packet has been routed by step 236, it is passed by step 238 to the CAL encoder 162, which
10 encodes the data packet in the CAL language.

As explained above with respect to FIG. 7A, each of the steps 234, 242 and 244 determines the route that data is transmitted via the MCU 112. As will be explained with respect to FIG. 7B, a data routing program 250 routes the data
15 packets depending on a number of factors including how the routed data is coded and/or the past handling of that data. As discussed above, each of the data packets is variously transmitted to the SCD 31 via one the first and second transmission media 19 and 16, and the passive bus 114. As
20 will be explained with respect to FIG. 7C, these data packets are variously transmitted over the first medium 19, the passive bus 114 and the second medium 16 to be stored in a corresponding one of the queues 113a, b or c, respectively. The temporary storage in the queues 113 permits later
25 processing when the SCD 31 is not processing a higher priority task. After the routing program 250 is called or initiated in

53

step 251, step 252 accesses one of the queues 113 to examine the data packet buffered there to determine its address and whether or not it is destined for the SCD 31. If addressed for the SCD 31, the data packet is processed in step 252 by
5 the SCD 31, and any response generated by the SCD 31 is routed in step 254 to the same input/output (I/O) corresponding to the interrupt by which the data was fed to the SCD 31. After step 254 has routed its data packet, the routing program moves to step 256, where an exit is made in step 256 from the
10 program 250.

If the data packet is not addressed for the SCD 31 as determined in step 252, the program determines in which queue 113 the examined data packet is buffered and, therefore, over
15 which of the first or second transmission media 19 or 16, or the passive bus 114 the data packet was transmitted to the SCD 31. Recall that in the illustrative embodiment shown in FIG. 4, that data transmitted over the first and second media 19 and 16 originated from the WAN 28 and the LAN 28,
20 respectively. If the data packet was transmitted from the LAN 30 as determined by step 258, step 259 sets the route to a default route, i.e., the present route set by the routing program 250. For example, if the present source from which the data packet was delivered to the SCD 31 is the LAN 30,
25 step 259 will set the routing destination for any response from the SCD 31 likewise to be the LAN 30. However if the present source for incoming data packets is the WAN 28, i.e.,

54

the SCD 31 is presently coupled to the central controller 18, step 259 sets the route to be to the WAN 28. When the central controller 18 stops talking to the SCD 31, step 259 sets the default route back to be to the LAN 30.

5

If the data packet is not addressed to the SCD 31, step 260 determines whether or not the data packet was transmitted to the SCD 31 from the WAN 28. If from the WAN 28, step 261 examines the address (HDLC) of the data packet. If HDLC
10 equals 1, step 262 routes the return signal to be to the LAN 30 and, if HDLC equals 2, step 263 sets the route for the return signal to be the passive bus 114.

However if the data packet is not addressed for the SCD
15 31 and is transmitted via the passive bus 114 to the SCD 114 as determined by step 260, step 265 examines the address borne by the data packet. If for the LAN 30 (HDLC equals 1), step 267 sets the route to be to the LAN 30. If routed for the WAN 28 (HDLC equals 2), step 269 sets the route to be to the WAN
20 28. After the routing has been completed, step 256 exits from the routing program 250.

The process, termed a hardware protocol, for transferring by program modules 150 a and b in FIG. 6 a single byte of data
25 to and from the passive bus 114 will now be explained in greater detail with respect to FIG. 4. The process is implemented in software executed by the MCU 112 to transfer a

55

single byte, bidirectionally through the data latch 124 between the passive bus 114 and the MCU 112. The program modules 150a and b operate the latch 124 in according to its hardware protocol, which is a part of its first, time critical layer. In other words, the hardware protocol operates in its time critical layer to cause each byte of the data packet to shift one at a time in either direction through the latch 124. At the first level, the program is oblivious to the information content of the data packet. On the other hand, the second, non-time critical layer operates to determine the information content of the packet without regard to how the packet is subdivided or otherwise transmitted to the bus 114.

To initiate the transfer in the first direction from the passive bus 114 to the MCU 112, the intelligent expansion device 122-1 places data on the passive bus 114, before transmitting an I/O write command via the bus 114 and the address line 121 to the address decoding circuit 126. In turn, the address decoding circuit 126 verifies the validity of the command's address, before applying a write control signal over line 119b to the latch 124, enabling the latch 124 to receive over line 117a from the bus 114 and store temporarily a single byte of data. Further, the address decoding circuit 126 transmits a data write signal via the interrupt line 115b to a maskable interrupt of the MCU 112 to alert it to a data byte stored in the latch 124 to be read,

56

and then releases the control line 119b, i.e., removes the data write signal from this line, whereby the data byte written to the latch 124 is latched for the MCU 112, i.e., the byte is ready to be read into the MCU 112. Now the
5 intelligent expansion device 122-1 resumes its task processing. The MCU 112 responds to the data write signal by raising the read control line 119a, i.e., applies a read control signal via line 119a to the latch 124, whereby the byte is transmitted to the MCU 112 via the data line 117b and
10 read therein. The MCU 112 releases the read control line 119a, before beginning to process the read byte of data.

To initiate the transfer of a data byte in a second opposite direction from the MCU 112 to the passive bus 114, the MCU 112 places the data byte on the data line 117b, and
15 then applies a write control signal via the control line 119a to the data latch 124, enabling the byte to be written into the latch 124. The write control line 119a is then released, allowing the latch 124 to latch the byte for the passive bus 114. The MCU 112 also transmits via the interrupt line 115a
20 an interrupt signal to alert the intelligent expansion device 122-1 to the presence of data in the latch 124. At this time, the MCU 112 may resume its independent data processing. The intelligent expansion device 122-1 responds to the interrupt signal by transmitting an I/O read signal with the appropriate
25 address via the address line 121 to the address circuit 126,

57

which responds to this read signal or request by transmitting a read signal via the control line 119b to the latch 124, enabling the latch 124 to write the data byte to the 8-bit data line 117a. The intelligent expansion device 122-1 now
5 reads the byte from the line 117a, before the address decoding circuit 126 releases the read control line 119b from the latch 124 and the device 122-1 processes the transferred data byte.

As discussed generally above with respect to FIG. 6, the MCU 112 has built in hardware interrupts, which are processed
10 on the highest priority basis by a priorities software module 275, which will be explained below with respect to FIG. 7C. The processing of the hardware interrupts will be given the highest priority over any other processing carried out by the MCU 112, even when a software event occurs at the same time.
15 The software is forced to wait until the processing of the hardware interrupt is finished before the software can proceed. The processing of the interrupts 133, 137 and 117b, to which the first transmission medium 19, the second medium 16 and the passive bus 114 are respectively connected, always
20 occurs first and is set at the highest priority levels based on the design of the MCU 112. As will be explained below, only after the hardware or high-priority interrupts have been processed, can the software or low-priority events be processed. The low-priority events are buffered so they can
25 easily be stored until the high-priority events can be processed. When no high priority hardware interrupts are

58

occurring, the priorities software module 275 may then handle the low-priority events. The high-priority hardware interrupt events are characterized by their very simple and fast operations. High priority events must occur very rapidly and efficiently to reduce the possibility of losing incoming data.

Thus in the context of this embodiment of the invention, the high priority events are the inputting of data to the SCD 31 and all other processing are assigned a lower priority. In particular, the inputting of data from the WAN 28 via its first transmission medium 19, from the LAN 30 from its second medium 16 and from the passive bus 114 to the corresponding interrupts 133, 137 and 117b are handled with the highest priority, higher than all other processing in this illustrative embodiment.

15

FIG. 7C represents both the hardware and software operation of the MCU 112. Steps 277 to 285 represent the operation of the hardware interrupts 137, 133 and 117b, while the remaining steps of Figure 7C are effected by the MCU 112 executing its software. When a packet of data appears at any one of these interrupts, step 277 will immediately interrupt the execution of all software. Figure 7C illustrates how the SCD 31 operates to set the priorities with which it carries out its various operations. Generally, the highest priority will be given to the detection of an interrupt, whereas a lower priority is given to operations carried out by the execution of software. As indicated by Figure 7C, in the

25

59

absence of an interrupt, the MCU 112 will execute step 282 or, if another software implemented process was previously interrupted by the detection in step 277 of an interrupt, to the step where the software program was interrupted. The SCD
5 31 will process the low priority, software executed tasks until a hardware interrupt occurs as illustrated by step 277, at which time the currently executed low priority program will be stopped, and step 278 detects whether there was a WAN
10 interrupt, i.e., a data packet was received via the first communication medium 19 at the WAN interrupt 133 of the MCU 112. If there was a WAN interrupt, a byte of the data appearing at this interrupt 133 is added in step 279 to its queue 113a. If no data appeared at the WAN interrupt 133, step 284 determines whether data appeared at the LAN interrupt
15 137, i.e., a data packet was received via the second medium 16 at the LAN interrupt 137, and, if yes, step 279 loads a byte of the data appearing at the LAN interrupt 137 into its queue 113c. If step 284 determines that the data did not appear at the LAN interrupt 137, step 285 determines whether or not data
20 appeared at the passive bus interrupt 117b. If yes, steps 279 adds a byte of that data to the bus queue 113b. If no, step 286 processes any other hardware interrupts. After the interrupts have been handled, the low priority program that was running before the hardware interrupts occurred is
25 restarted.

If no hardware interrupts occur as detected by step 277,

60

the priorities software module 275 moves to step 282, which determines whether there are any data bytes in the queues 113a, b and c. If there is data in the queues 113 for processing, step 283 performs a variety of relatively low
5 priority processing, e.g., checking the CRC code for errors in the data packet by step 154a, decoding the CAL formatted data in step 156 or carrying out a routed scheduling process in the task scheduler 160 as more fully explained with respect to FIGS. 7A and B. If there are no data packets in any of the
10 queues 113 awaiting low level processing as determined in step 282, step 287 determines whether there are any miscellaneous processing to do, such as time keeping, voltage sampling etc.

After step 288 has been completed or step 287 determines that there is no further miscellaneous processing to be carried
15 out, step 280 exits the priorities software module 275.

The scaleable expansion protocol used in the SCD 31 of the invention is unique because it allows the automatic detection of an intelligent expansion device 122-1 and/or
20 device error/failure. As shown in FIG. 8A, the intelligent expansion device 122-1 attached to the passive bus 114 uses a scaleable expansion protocol to periodically communicate with the SCD 31 and requires the SCD 31 to respond back acknowledging the presence of the intelligent expansion device
25 122-1. If an intelligent device 122-1 attached to the passive bus 114 fails, the MCU 112 will recognize this condition and

61

will begin the error recovery or correction procedure. The error recovery procedure will allow partial functionality without total system failure.

Generally, the "intelligent" device 122-1 sends a hand-shake packet out on the data bus 114. The hand-shake packet may include either of two single bytes representing FE (254) of FF (255). The FE byte serves as a first synchronization hand-shake, while the FF byte serves as a second synchronization hand-shake. The use of these two hand-shake bytes allows the SCD 31 to synchronize its operation with that of the inserted intelligent device 122-1 if needed. The intelligent device 122-1 will send the hand-shake packet out on the data bus 114 every 5-10 seconds depending on the current activity of a CPU not shown) associated the bus 114.

If an "intelligent" device 122-1 is available and operating correctly, that device 122-1 will initiate the hand-shake sequence by alternating the single byte hand-shake packet starting with FE, then FF, and so on. If no valid hand-shake sequence is received by the SCD 31, the intelligent expansion device 122-1 is considered not available. If the intelligent expansion device 122-1 was operational and has ceased to communicate, that condition is flagged internally as an error and the SCD 31 attempts to operate in a reduced capability mode as will be explained with respect to FIG. 8A.

62

The intelligent device 122-1, as shown and described generally above with respect to FIG. 5, is a combination of a CPU, a memory and additional hardware (not shown in the drawings) that is controlled by a device driver (special software) to allow it to communicate with the SCD 31. The driver software is stored in the aforementioned memory and executed by the CPU of the intelligent expansion device 122-1.

A further program module identified by the numeral 300 in FIG. 8A, termed a scaleable expansion protocol, is also stored within this memory and executed by this CPU. The scaleable expansion protocol 300 facilitates the automatic detection of the insertion an intelligent expansion device 122-1 within any one of the slots 120-1 to n of the passive bus 114. The expansion protocol program 300 is periodically executed to transmit a hand-shake packet from an intelligent expansion device 122-1 via the buffered latch 124 to the MCU 112. Such hand-shake messages are repeatedly sent at regular intervals to signal the presence or absence of an intelligent expansion device 122-1 on the passive bus 114 and to indicate whether the intelligent expansion device 122-1 is operating correctly and, if not, to initiate procedures to restore its operability. The hand-shake message requires an inserted intelligent expansion device 122-1 to start the hand-shake sequence followed by the MCU 112 responding in one illustrative embodiment of this invention with a reply

63

message, which acknowledges the operational status of the intelligent expansion device 122-1. If an inserted intelligent expansion device 122-1 fails, the expansion protocol 300 will recognize this condition and will initiate its error recovery or correction procedure. The error recovery procedure will at least permit partial operation of the MCU 112.

With reference now to FIG. 8A, there is shown a flow diagram of the expansion protocol which is generally referenced by the numeral 300 and will now be explained in terms of its detailed steps. In step 302, the expansion protocol 300 is called. This protocol or program 300 is an application program which runs on the processor or CPU (not shown) of the intelligent expansion device 122-1. Upon insertion of the intelligent expansion device 122-1 on the passive bus 114, the hand-shake packet described above is repeatedly transmitted in step 304 out on the passive bus 114. This hand-shake packet will be sent in step 304 every 5-10 seconds and periodically changes its form between its value of FE and its value of FF, whereby successively transmitted packets are of different values. For example, the first value of the packet is FE and the next value of the packet is FF. If an intelligent expansion device 122-1 is present on the passive bus 114, the device 122-1 upon coupling to the passive bus 14 will generate in step 304 a sequence of data packet to

64

let the MCU 112 know of its presence on the bus 114. In an alternative embodiment of this invention, the MCU 112 may respond with an acknowledgement packet. Next, step 305 will listen for the return hand shake packet of the intelligent device 122-1 and, if present on the passive bus 114, the MCU 112 will receive in step 306 the return packet. In an alternative embodiment of this invention, the MCU 112 may respond to the receipt of the data packet by generating and transmitting an acknowledgement signal back to the intelligent expansion device 122-1. If an intelligent expansion device 122-1 is present on the passive bus 114, the MCU 112 then, in step 310, checks a table maintained in its system memory 138 for the presence of a flag, which indicates the receipt by the MCU 112 of a previous data packet and that another intelligent expansion device had been inserted previously on the passive bus 114. If there is no flag in the memory 138, step 310 provides a yes message that the received packet indicates that a first, new intelligent expansion has been inserted into the passive bus 114. Responding to that yes message, step 312 sets a flag into the aforementioned table in memory 138, before step 318 exits the scaleable expansion protocol program 300.

In an alternative embodiment of this invention, the repeated interrogating signals could be generated by the MCU 112 and a response generated by the intelligent expansion

65

device 122-1 when it is inserted into the passive bus 114.

On the other hand if step 310 sends a no message indicating that another intelligent expansion device 122-1 had been previously inserted on the passive bus 114, step 314
5 generates and stores a signal in the noted table indicating that the previously inserted intelligent expansion device was still working, i.e., was still regularly transmitting its periodic data packets. Then step 316 uses the alternating packets, first an FE packet and then an FF packet, to
10 synchronize receipt of a data message or other functions carried on in the MCU 112. Then step 318 exits from the protocol expansion program 300.

If step 305 does not detect the return of a hand shake packet as would be generated if an intelligent expansion
15 device 122-1 has been coupled to the passive bus 114, the expansion protocol program 300 moves to step 320, which examines that table built in step 312 to determine whether an intelligent expansion device had been previously coupled to the passive bus 114. If not, the program 300 proceeds to step
20 318, where an exit is made. On the other hand if the prior coupling of an intelligent expansion device 122-1 is found, step 322 recognizes the presence on the passive bus 114 of an intelligent expansion device 122-1 that is no longer operating properly and generates a flag which is indicative of that
25 device's failure. Step 324 logs that flag as an error in the

66

system memory 138. Next, step 326 initiates an error recovery procedure, whereby that indication set into the table of the system memory 138 of the presence of a particular expansion device 122 on the passive bus 114 is erased. Thus, the SCD 31
5 can continue to operate as if the non-functional expansion had never been coupled to the bus 114.

In a preferred embodiment of this invention as shown in FIGS. 5 and 8B, the SCD 31 utilizes the passive bus 114, which
10 is configured in accordance with an Industry Standard Architecture (ISA) IEEE-P996 or PC/104 IEEE-P996.1 to communicate between itself and the expansion devices 122 that are coupled to the passive bus 114. Such devices 122 may illustratively comprise CPU cards, video cards, disk drive
15 controllers, etc. In another illustrative embodiment of this invention, the passive bus 114 may take the form of Intel's Universal Serial Bus, Peripheral Component Interconnect (PCI) IEEE-P1386.1, EIA 232 or EIA 485. As shown in FIG. 5, the expansion devices 122-1 to n are associated respectively with
20 data controllers 123-1 to n. The data controller 123-1 associated with the intelligent expansion device 122-1 is the master device (bus controller) controlling the data flow between the other data controllers 123-2 to n and subsequently the data flow between the passive expansion devices 122-2 to n
25 that are residing in the passive expansion slots 120-2 to n. The term "intelligent" indicates that "master" or "controlling" functions are implemented by the intelligent

67

expansion device 122-1. Likewise, the term "passive" indicates that a "slave" or "controlled" role is carried out by the passive expansion devices 122-2 to n. The passive bus 114 operates as a "master" device and a "slave" device over which data can be bi-directionally requested from various devices 122 coupled to the passive bus 114 to facilitate the bi-directional transmission of data over that bus 114.. The SCD 31 in the preferred embodiment utilizing IEEE-P996 would be a passive device in the sense explained above and be similar to the combination of a passive expansion device 122-2 and its data controller 123-2.

As particularly shown in FIG. 8B, the data controllers 123-1 to n are interconnected by the passive data bus 114 and the other illustrated buses to permit at least the following data transmissions between the expansion devices 122-1 to n:

- 1) data transmitted from the intelligent expansion device 122-1 to any of the passive expansion devices 122-2 to n,
- 2) data transmitted from one of the passive expansion devices 122-2 to n to the intelligent expansion device 122-1, and
- 3) data transmitted from one of the passive expansion devices 122-2 to n to another of these passive expansion devices.

To facilitate such data transfers, an address bus 125 interconnects each of the data controllers 123-1 to n whereby the address for the transmitted data may be forwarded to a particular data controller 123. A plurality of device read/write control lines 116 interconnect the expansion

68

devices 122-1 to n to control either a read or a write data transmission with respect to a particular expansion device 122. Further, there is a set of device interrupt lines 127-1 to n, which connect respectively that data controller 123-1, which is associated with the intelligent expansion device 122-1, to one of the data controllers 123-2 to n, which are associated with the passive expansion devices 122-2 to n, respectively. As will be explained below, the interrupt lines 127 facilitate the transmission of data from one of the passive expansion devices 122-2 to n to the intelligent expansion device 122-1.

When it is desired to transmit data from an intelligent expansion device 122-1 to a selected one of the passive expansion devices 122-2 to n, the data controller 123-1 associated with the expansion device 122-1 places the desired data on the passive bus 114, then places a "write status" on the device read/write control line 116 before writing the system address of the addressed (selected) one of the passive devices 122-2 to n onto the address bus 125. The data controllers 123-2 to n recognize their own unique range of addresses. The data controller 123 for that one of the passive expansion devices 122-2 to n that corresponds with the unique address(es) then examines the status of its device read/write control line(s) 116 as shown in FIG. 8B to determine the direction of the data transmission. For this data transmission, the control line(s) 116 controls the

BEST AVAILABLE COPY

69

intelligent expansion device 122-1 to write data to the corresponding passive expansion device. After detecting its unique address(es) and the status of the read/write control lines 116, the data controller 123 associated with the uniquely addressed passive expansion device then latches the data that is placed on the passive data bus 114 and then stores the transmitted data to a memory address within the passive expansion device 122-2 to n which corresponds to the unique address(es) present on the address bus 125. At this point, the transfer of one portion of the data between the intelligent expansion device 122-1 and the addressed one of the passive devices 122-2 to n is complete.

When it is desired to transmit data from one of the passive expansion devices 122-2 to n to the intelligent expansion device 122-1, a data request sequence is initiated by that particular data controller 123 associated with the corresponding passive expansion device 122-2 to n that needs to transfer data to the intelligent expansion device 122-1 by activating one of the device interrupt lines 127-1 to n. As shown in FIG. 8B, the device interrupt lines 127 extend between the data controller 123-2 to n and the data controller 123-1. Each data controller 123-2 to n is assigned a device interrupt line 127 which may or may not be unique with respect to the other data controllers 123. Once the corresponding device interrupt line 127 is activated, the data controller 123-1 recognizes that one of the data controller(s) 122-2 to n

70

from which the request came. The data controller 123-1 then places a "read status" on the corresponding one of the device read/write control line(s) 127 and writes the system address of the requesting passive expansion device 122-2 to n onto the address bus 125. The one data controller 123 for the requesting passive expansion device 122-2 to n recognizes it's own unique range of addresses. The data controller 123 for the passive expansion device 122 that corresponds with the unique address(es) then examines the status of its device read/write control line(s) 116 as shown in FIG. 8B to determine the direction of the data transmission. For this case, the read/write control line(s) 116 indicate that the intelligent expansion device 122-1 needed to read data from the corresponding passive expansion device. After detecting the unique address(es) of the passive expansion device that is transmitting data and the read/write status appearing on the control lines 116, the data controller for the uniquely addressed passive expansion device inserts on the data bus 114 the data intended for the intelligent expansion device 122-1. The data controller 123-1 then latches the data from the passive data bus 114 and transfers it to the intelligent expansion device 122-1. At this point, the transfer of one portion of data from the one transmitting passive expansion device 122-2 through 122-n to the intelligent expansion device 122-1 is complete.

In order to transmit data from one of the passive

BEST AVAILABLE COPY

7/

expansion devices 122-2 to n to another, each of the passive expansion devices involved in this data transfer must communicate through the intelligent expansion device 122-1. A sequence of steps to carry out this transfer of data can be as follows: First, the one passive expansion device 122-2 to n from which data is to be transferred 122-2 performs a passive to intelligent expansion device data transfer as described above. Included in the data transferred from the transmitting passive expansion device 122-2 to n is also a request for the data receiving intelligent expansion device 122-1 to forward the received data onward to a particular one of the receiving passive expansion devices 122-2 to n. The intelligent expansion device 122-1 then transfers the data received from the transmitting passive expansion device 122-2 to n as described above. This two step process can continue until the desired data is transferred from one of the passive expansion devices 122-2 to n to another of these devices and vice-versa.

20 The CEBus protocol used by the PLC or LAN controller 32 to transmit data signals over the second transmission medium in the form of the PLC or secondary power line 16, did not provide a robust method for transmitting data over the first medium in the form of a noisy communication medium such as the RF link 19. As explained above, data packets transmitted to 25 the LAN 30 over the secondary power line 16 are structured in accordance with the CEBus Standard EIA-600. As will be

BEST AVAILABLE COPY

72

explained below with respect to FIG. 9, such CEBus data packets are modified to permit the ready detection of transmission errors as would be imposed on the data packet as it is transmitted over the noisy RF link 19. In that
5 illustrative embodiment of this invention as shown in FIG. 9, such a modified or error checking data packet as identified by the numeral 340. The structure of this data packet 340 was designed as will be explained to allow the reliable detection of errors. This illustrative structure facilitate the
10 detection but not the correction of noise errors.

Generally, the design of the error checking data packet 340 for transmission over the RF link 19 between the RF controller 34 and the central controller 18, allows its associated software to detect if an error has occurred in each
15 data packet. If an error is detected, the data packet 340 is discarded and no action is taken. The burden of confirming the transmission of a data packet over the RF link 19 and its reception by the central controller 18 is placed on the central controller 18.

20 With reference now to FIG. 9, the structure of an error checking data message 340 comprises a series of variable and fixed length parts 340 a-f, each part serving a different function as will be explained. This packet structure is illustratively defined as HDLC (ISO/IEC 3309:1993(E)). The
25 first part comprises a unique flag byte 340a. This unique flag byte notifies the receiving device, i.e., the central

BEST AVAILABLE COPY

23

controller 18, that a valid packet is about to be received and also serve to synchronize the reception of the message 340. The flag byte 340a will comprise 7E to form in effect the synchronization header as explained above. Once the central
5 controller 18 is synchronized to the stream of data coming in, the controller 18 will begin receiving the entire error checking packet 340. A second part 340b comprises a single, address byte of data which indicates the route that the message 340 will take once it reaches the SCD 31. The address
10 byte 340 may contain a 1 which specifies that the message will be handled by the SCD 31 or the address byte 340b may contain a 2 which specifies that the packet will be sent to the passive bus 114. The third part 340c comprises a single, control byte that is currently not used and filled with 0.
15 The fourth part 340d is of variable length and is the CEBus data packet which contains the CAL message. The fifth part 340e includes two bytes (a total of 16 bits), which permit an integrity check of the received bytes to be made. The sixth and final part 340f comprises a unique flag byte that
20 signifies that the packet is finished. In an illustrative embodiment of this invention, the well known cyclic redundancy check (CRC) performs a mathematical process of checking the integrity of the series of bits that are subcomponents of a byte to determine if they have arrived over the noisy RF line
25 19 unaltered. When the message 340 is received, the packet address byte 340b, the control byte 340c, and the CEBus data packet part 340d are checked using the fifth part 340e. If

74

the above criteria are met, the message 340 is accepted as valid and sent on to the non-timing critical level software for further processing.

5 The CRC checking described above provides error checking of the data transmitted, but does not provide for data correction. Data recovery or correction of the message 340 is facilitated by their numbering, i.e., to provide a level of error recovery over noisy communication media, such as the RF
10 link 19. Each packet 340 within each group or block of packets is given a number indicating its position in the block. By using the numbers in conjunction with the packets, bad or missing packets are detected and recovered with the next block of data as will be explained below with respect to
15 FIG. 10. This recovery or correction is particularly adapted for use with transponders 22, which may illustratively take the form of those meters made by General Electric and known as "Phase 3" meters.

Each block of packets has the following three important
20 characteristics that allow for error recovery: 1) the first packet 340 of a block is marked by setting the most-significant bit to a "1"; 2) each packet 340 contains a number indicating how many packets remain in the block; and 3) each packet 340 must pass the 16-bit CRC for data
25 integrity, as explained above.

-75-

As each packet 340 is received, the packet is placed in a row of a TABLE that corresponds with the number associated with the packet. That TABLE is identified by the letter A as illustrated below and may be formed within the MCU 112. As the row of the TABLE A is filled with a packet 340, a flag is set to show that this packet 340 is not needed if the block is asked for again. Once TABLE A is filled with valid packets, the block is considered complete and can be processed.

TABLE A:

Order packet was Received	Packet Number in Hexadecimal	Packet Number in Binary
1	84	1000 0100
2	03	0000 0011
3	02	0000 0010
4	01	0000 0001
5	00	0000 0000

10

The packet number "84" is actually packet "04" with the left-most bit set to a "1" changing its value to "84". This hexadecimal value greater than "80" indicates that it is the first packet. In order to interpret the packets 280 correctly, the packets must be decoded in sequence starting with packet "84" down to packet "00". The right-most seven bits of the hexadecimal packet number are used as an index

-76-

into a TABLE identified by the letter B for storage of the packets as shown in below.

5 TABLE B:

Packet Number in Hexadecimal	Flag to show packet has arrived	CEBus Packet
4	YES	Packet Contents
3	YES	Packet Contents
2	YES	Packet Contents
1	YES	Packet Contents
0	YES	Packet Contents

If any of the flags set in TABLE B are not set to 'YES', the block of packets will be asked again to fill only the empty slots in the TABLE B. According to statistics and a
 10 random distribution of errors, the technique above will work even in a noisy communication environment.

In FIG. 10, a flow chart of that error recovering procedure 400 applied to the error checking data packets 340 transmitted via the noisy RF link 19 and its controller 34, is
 15 provided. Initially in step 402, TABLE B is cleared in preparation to receive a new block of CEBus packets 340. In particular, step 402 clears the table flag that represents

??

whether a particular packet has been received, by setting its flag to "NO". Then step 404 sends a CEBus packet 340 containing a message to retrieve a block of data from that device connected to the RF link 19 and included within the WAN 5 28, e.g., the central controller 18. Next step 406 determines whether or not a packet has arrived in response to the previous request. If a packet has been receive as determined in step 406, then step 408 uses the CRC error checking procedure described above to determine whether or not the 10 received packet is valid or not. If the received packet is invalid, step 418 discards the invalid packet. If valid as determined by step 408, step 412 inserts the valid data packet into its proper place in the packet TABLE B and sets its received flag to "YES". Then step 414 checks whether TABLE B 15 has been completely filled with CEBus packets by scanning TABLE B from the first packet numbered 4 to the last numbered 0 and, if all of the flags are set to "YES", then TABLE B is completely filled. If TABLE B is not completely filled as determined by step 414, the data error recovery program 400 20 loops back to step 404 to again call the next data packet to be processed as described above in steps 406, 408 and 412, before step 414 again checks whether TABLE B has been filled. If not, the recovery program 400 continues to loop through steps 406, 408 and 412, until step 414 determines that the 25 TABLE is filled, thus completing the program 400 when TABLE B is filled, step 416 exits the program 400.

Each time a request for a block of data is made in step

BEST AVAILABLE CC

78

404, step 404 resets a clock to time that period in which a data packet should be received. If step 406 determines that a packet has not been received, the program 400 moves to step 420, which determines whether the period set for response in
5 step 404 has expired. As shown in FIG. 10, the program 400 continues to loop through steps 406 and 420 until the response time clock has expired. At that time, step 420 increments an internal counter. After the counter has been incremented, the program 400 returns to step 406 to initiate the transmissions
10 of further message requests for a block of data. Then step 422 tests whether the count stored in the internal counter has reached a predetermined number and, if not, the program 400 loops through steps 404, 406 and 420 to repeatedly send requests for a data block. When step 422 indicates that the
15 number of no-responses reaches the predetermined number, step 424 logs an error condition and internally flags the central controller 18 that a block of packets could not be retrieved, before an exit is made in step 416 from the program 400.

It will thus be seen that the objects set forth above,
20 among those made apparent from the preceding description, are efficiently attained and, since certain changes may be made in the constructions set forth without departing from the spirit and scope of the invention, it is intended that all matter contained in the above description and shown in the
25 accompanying drawings shall be interpreted as illustrative and not in a limiting sense.

79

It is also to be understood that the following claims are intended to cover all of the generic and specific features of the invention herein described and all statements of the scope of the invention which, as a matter of language, might be said
5 to fall therebetween.

THIS PAGE BLANK (USPTO)

THIS PAGE BLANK (USPTO)

WHAT IS CLAIMED IS:

1. An expandable interface for coordinating the transmission of data between a first network and a second network, said interface comprising:

- a) a bus for releasably receiving selected of a plurality of expansion devices;
- b) a programmable processor being connected to receive and transmit a bidirectional flow of data to and from each of the first and second networks and to and from said bus;
- c) a latch being connected to receive and transmit a bidirectional flow of data to and from each of said bus and said processor; and
- d) said processor being multitasked programmed to perform at least first and second tasks, said first task controls the flow of data between the first and second networks via said interface, between the first network and said bus and between the second network and said bus, said second task controlling said latch to receive and transmit data to and from said processor.

2. An expandable interface for coordinating the transmission of data between a first network and a second network, said interface comprising:

- a) a plurality of expansion devices;
- b) a bus for releasably receiving selected of a plurality of expansion devices;
- c) a first programmable processor being connected to

81

receive and transmit a bidirectional flow of data to and from each of the first and second networks and to and from said bus;

d) a latch being connected to receive and transmit a bidirectional flow of data to and from each of said bus and said processor;

e) said plurality of expansion devices including at least one smart expansion device, said smart expansion device comprising a second programmable processor; and

f) said first and second programmable processors each operating asynchronously with respect to each other, said first and second programmable processors being programmed to operate said latch to effect a bidirectional transfer of data via said latch in synchronism between said first and second programmable processors.

3. A computer system adapted to be expanded comprising:

a) a first programmable processor including at least one port;

b) an expansion coupler adapted to be connected in a data transmitting relation to selected of a plurality of expansion devices;

c) at least one expansion device that is adapted to be connected to said coupler and includes a second programmable processor;

d) a first buffer connected to said one port; and

e) a second buffer connected to said expansion coupler;

f) said first programmable processor being multitasked

82

programmed to load first data destined for said selected of said plurality of expansion devices to said first buffer, to place a first data available message on said coupler and to respond to a second data available message to download second data destined for said first processor from said second buffer via said coupler and said one port to said first programmable processor;

g) said second programmable processor being multitasked programmed to transmit said second data via said coupler to said second buffer, to place said second data available message on said one port and to respond to said first data available message to download from said first buffer said first data onto said coupler.

4. The expandable computer system as claimed in claim 3, wherein said first programmable processor further comprises second and third ports.

5. The expandable computer system as claimed in claim 4, wherein said first programmable processor is connected respectively via said second and third ports to first and second paths of bidirectionally flowing data.

6. The expandable computer system as claimed in claim 3, wherein said first and second programmable processors operate asynchronously of each other.

7. The expandable computer system as claimed in claim 3, wherein said first programmable processor is further programmed to receive data appearing at any of said one, second and third ports and to route said received data to

BEST AVAILABLE COPY

83

another one of said one, second and third ports.

8. The expandable computer system as claimed in claim 3, wherein there is further included a second expansion device that is connected to said expandable coupler, and a data controller is included in one of said one and second expansion devices to control the flow of data to and from each of said one and said second data expansion devices.

9. The expandable computer system as claimed in claim 8, wherein said data controller is included within said first expansion device.

10. The expandable computer system as claimed in claim 8, wherein said data controller is included within said second expansion device.

11. An expandable interface for coordinating the transmission of data between first and second network systems, said interface comprising:

a) a coupler to connect selected of a plurality of expansion devices;

b) at least one expansion device adapted to be connected to said coupler and including a first programmable processor; and

c) a second programmable processor comprising at least first, second and third ports connected respectively to said first network system, said second network system and said coupler, said second programmable processor being multitasked programmed to perform at least first, second and third tasks, said first task to receive data at any one of said first,

BEST AVAILABLE COPY

84

second and third ports, said second task to select another one of said first, second and third ports, and said third task to route data to said selected other port.

12. The interface as claimed in claim 11, wherein said second task further includes programming to determine at which one port of said first, second and third ports said data was received and to select said other port dependent at least in part on said determined one port.

13. The interface as claimed in claim 11, wherein data transmitted over any of said first, second and third operating networks includes an address, said second task is programmed to detect said address and to select said other port dependent at least in part on said detected address.

14. The interface as claimed in claim 11, wherein said first operating network includes a central controller, said second task is programmed to detect the status of the central controller in terms of whether it is connected on-line with said second processor and to select said other port dependent at least in part on said on-line status of the central controller.

15. The interface as claimed in claim 11, wherein said first, second and third ports of said second processor comprise respectively first, second and third interrupts.

16. The interface as claimed in claim 15, wherein there is further included first, second and third buffers connected respectively with said first, second and third interrupts.

17. The interface as claimed in claim 16, wherein said

BEST AVAILABLE COPY

85

second processor is further multitasked programmed to perform a fourth task, said fourth task to detect the presence of data at any one of said first, second and third interrupts to suspend the processing of any of said first, second and third tasks and to transfer a unit of said data appearing at said one port to its corresponding buffer.

18. The interface as claimed in claim 17, wherein said fourth task is programmed to detect the receiving of data at any one of said first, second and third interrupts and, upon the detecting of the receiving of data, to suspend the execution of said first, second or third tasks and to transfer a unit of said received one of data at said one port to its corresponding one of said first, second and third buffers.

19. The interface as claimed in claim 18, wherein said fourth task is responsive to the loading of said unit data into its one corresponding buffer to resume executing of said one suspended task.

20. The interface as claimed in claim 11, wherein data is transmitted over one of said first network, said second network and said coupler as a packet comprised of a selected number of data units encoded with a wrap in accordance with a protocol unique to said corresponding one of said first network, said second network and said coupler over which said data packet is to be transmitted, and there is further included first, second and third buffers connected respectively with said first, second and third ports.

21. The interface as claimed in claim 20, wherein said

BEST AVAILABLE COPY

86

first task detects the presence of one unit data of its packet at one of said first, second and third ports to receive and store said one unit data into said buffer connected to said one port.

22. The interface as claimed in claim 21, wherein said first task further detects the presence of one unit data at said one port to provide a signal indicating which one of said first network, said second network and said coupler said received unit data was transmitted over and to place said indicating signal with said data unit in said buffer connected to said one port.

23. The interface as claimed in claim 22, wherein said second processor is further multitasked programmed to perform a fourth task that is responsive to said indicating signal to transfer and assemble each unit data of a particular packet and its indicating signal into a queue.

24. The interface as claimed in claim 23, wherein there is further included a task memory, said fourth task responsive to the collection and reassembly of the selected number of data units into said data packet to transfer said reassembled data packet from said queue into said task memory, wherein said second task processes said reassembled data packet to select said port to which said data is to be routed.

25. The interface as claimed in claim 24, wherein said fourth and third tasks respectively accesses and routes the next available data packet from said task memory to said selected port.

BEST AVAILABLE COPY
BEST AVAILABLE COPY

87

26. The interface as claimed in claim 24, wherein said second processor is multitasked programmed to perform a fifth task, said fifth task is executed to translate a complete, reassembled data packet by stripping away said wrap to provide said selected number of raw data units and to store said raw data units in said task memory.

27. The interface as claimed in claim 26, wherein said second and third tasks are executed to access and route a next available selected number of raw data units in said task memory to said selected port, said fifth task being executed by said second programmable processor to encode said selected number of raw data units with a wrap defined by the protocol of that one of said first network, said second network and said coupler over which said encoded data packet is to be transmitted.

28. The interface as claimed in claim 27, wherein said second programmable processor is further multitasked programmed to perform a sixth task to transfer said encoded data packet into a second queue and to transfer one data unit at a time from said second queue to said buffer connected to said port that corresponds with said network or coupler over which the encoded data packet is to be transmitted.

29. The interface as claimed in claim 11, wherein there is included first and second buffers interconnected between said first and second programmable processors.

30. The interface as claimed in claim 29, wherein said first and second buffers are coupled to said third port of

BEST AVAILABLE COPY

88

said second programmable processor and to said coupler.

31. The interface as claimed in claim 11, wherein at least two of said first network, said second network, said first programmable processor and said second processor operate asynchronously with each other.

32. The interface as claimed in claim 11, wherein each of said first network, said second network, said first programmable processor and said second process operate asynchronously with each other.

33. The interface as claimed in claim 11, wherein said first programmable processor is multitasked programmed to perform at least fourth and fifth tasks, said fourth task to control the flow of data between said one expansion device and said third port, said fifth task to carry out an expansion process.

34. The interface as claimed in claim 33, wherein a second expansion device is connected to said coupler, said fourth task is executed to further control the flow of data between said second expansion device and said coupler.

35. The interface as claimed in claim 11, wherein said second processor further includes a priority setter for determining the priority with which said first, second and third tasks are carried out.

36. The interface as claimed in claim 35, wherein said priority setter is implemented by a fourth task programmed into said second programmable processor, said fourth task is executed to set the priority with which said first, second and

BEST AVAILABLE COPY

89

third tasks are carried out.

37. The interface as claimed in claim 36, wherein said fourth task responds to the presence of data at any one of said first, second and third ports to execute as the highest priority said first task to receive said data at said one port.

38. The interface as claimed in claim 35, wherein said priority setter is implemented by configuring said first, second and third ports respectfully as first, second and third hardware interrupts.

39. The method of expanding and operating an expanded interface device, the interface device comprising a first processor with first, second and third ports, the first and second ports being connected respectively to first and second networks, the third port being connected to a coupler adapted to be connected to selected of a plurality of expansion devices, said method comprising the steps of:

a) connecting to the coupler at least one expansion device that includes a second processor;

b) detecting the presence of data at any one of the first, second and third ports;

c) receiving the data appearing at the one port;

d) selecting another of the first, second and third ports; and

e) routing the received data to the selected other port.

40. The method of operating an interface device as claimed in claim 39, wherein there is further included the

BEST AVAILABLE COPY

90

step of operating the one expansion device to transmit data to and from the third port.

41. The method of operating an interface device as claimed in claim 40, wherein there is further included the step of connecting a second expansion device to the coupler and operating the first mentioned and second expansion devices to control the transmission of data between the coupler and the first mentioned and second expansion devices in a coordinated manner.

42. The method of operating an interface device as claimed in claim 39, wherein step d) determines at which one of the first, second and third ports the data was received and selects the one port based at least in part on which one of the ports received the data.

43. The method of operating an interface device as claimed in claim 39, wherein an address is embedded into data that is indicative of the destination of that data and there is further included the step of detecting the address of the data received at the one port and said step d) of selecting selects the one port based at least in part on the determined data address.

44. The method of operating an interface device as claimed in claim 39, wherein the first network includes a central controller, and there is further included a step of determining the on-line status of the central controller and step d) of selecting selects the one port based at least in part of the on-line status of the central controller.

BEST AVAILABLE COPY

91

45. The method of operating an interface device as claimed in claim 39, wherein said method further comprises the step of periodically transmitting an inquiry signal from the interface to test the presence or absence of the expansion device connected to the coupler.

46. An expandable interface for coordinating the transmission of data between first and second network systems, said interface comprising:

a) a coupler adapted to connect selected of a plurality of expansion devices:

b) at least one expansion device adapted to be connected to said coupler and including a first programmable processor; and

c) a second programmable processor comprising at least first, second and third ports connected respectively to said first network system, said second network system and said coupler, said second programmable processor being multitasked to perform at least first, second, third and fourth tasks, each of said first, second and third tasks being assigned a priority and a corresponding command for its execution, said first task being executable to receive data appearing at any one of said first, second and third ports, said second task being executable to process any received data, said third task being executable to transmit data via at least one of said first, second and third ports, said fourth task responsive to one of said task calls to compare said assigned priority of said called task with that of any currently executed task and,

BEST AVAILABLE COPY

92

if said priority of said called task is higher than that of said currently executed task, suspending and currently executed task and higher priority executing said called task.

47. The interface as claimed in claim 46, wherein said first task is assigned the highest priority.

48. The interface as claimed in claim 47, wherein said third task is assigned the lowest priority.

49. The interface as claimed in claim 46, wherein a detector is responsive to the appearance of data at any of said first, second and third ports for providing and call for said first task.

50. The method of expanding and operating an expanded interface device, the interface device comprising a first processor with first, second and third ports, the first and second ports being connected respectively to first and second networks, the third port being connected to a coupler adapted to be connected to selected of a plurality of expansion devices, said method comprising the steps of:

- a) connecting to the coupler at least one expansion device that includes a second processor;
- b) receiving data at any one of the first, second and third ports;
- c) processing the received data;
- d) transmitting data via at least one of the first, second and third ports;
- e) assigning a priority to each of said receiving step

BEST AVAILABLE COPY

93

b), said processing step c) and said transmitting step d); and
f) placing a call for one of said receiving step b), said processing step c) and said transmitting step d); and
g) comparing said priority assigned to said step corresponding to said one call with said priority of said step currently being carried out and, if higher, carrying out said called step.

51. The method of operating an interface as claimed in claim 50, wherein said assigning step e) assigns the highest priority to said step b) of receiving data.

52. The method of operating an interface as claimed in claim 51, wherein said assigning step e) assigns the lowest priority to said step of step d) of transmitting data.

53. The method of expanding and operating an expandable computer system, the expandable computer system including a first processor with at least one port, a coupler adapted to be connected in a data transmitting relation to selected of a plurality of expansion devices, a first buffer connected to the first port and a second buffer connected to the coupler, said method comprising the steps of:

a) connecting to the coupler at least one expansion device that includes a second processor;

b) operating the first processor to transmit first data destined for the selected of the expansion devices via its one port to the first buffer and to place a first data available message on the coupler;

BEST AVAILABLE COPY

94

c) operating the second processor to transmit second data destined for the first processor via the coupler to the second buffer and to place a second data available message on the one port;

d) operating the first processor to respond to the second data available message to download from the second buffer the second data onto the coupler; and

e) operating the second processor to respond to the first data available message to download from the first buffer the first data onto the coupler.

54. The method of expanding and operating as claimed in claim 53, wherein said steps b) and d) of operating the first processor are carried out in a first time domain, and said steps c) and e) of operating the second processor are carried out in a second time domain that is independent of the first time domain.

55. The method of expanding and operating as claimed in claim 54, wherein said steps b) and d) of operating the first processor are carried out asynchronously with respect to said steps c) and e) of operating the second processor.

56. A computer system adapted to be expanded comprising:

a) a first programmable processor including at least one port;

b) an expansion bus adapted to receive in a data transmitting relationship selected of a plurality of expansion devices;

c) at least first and second expansion devices, each of

BEST AVAILABLE COPY

95

said first and second expansion devices is adapted to be connected in a data transmitting relationship with said expansion bus;

d) at least one buffer interconnected between said one port and said expansion bus;

e) a second programmable processor included within one of said first and second expansion devices, said second programmable processor being multitasked programmed to transmit data destined for said first programmable processor via said expandable bus to said one buffer and further to place a data available message on said one port of said first programmable processor; and

f) a data traffic controller included within one of said first and second expansion devices to control the flow of data from each of said first and second expansion devices onto said expansion bus.

57. An expandable computer system as claimed in claim 56, wherein said second programmable processor is included with said first expansion device and said data traffic controller is included within said second expansion device.

58. An expandable computer system as claimed in claim 56, wherein both of said second programmable processor and said data traffic controller are included within said first expansion device.

59. An expandable computer system as claimed in claim 56, wherein said data traffic controller is included within said first expansion device, and there are further included first

BEST AVAILABLE COPY

96

and second data transmitters included respectively within said first and second expansion devices for transmitting data onto said data expansion bus.

60. An expandable computer system as claimed in claim 59, where said second data transmitter applies a data ready message to said data traffic controller included within said first expansion device, said data traffic controller applying at a particular time a data transmit message to said second data transmitter, whereby said second data transmitter transmits its data onto said expansion bus.

61. An expandable computer system as claimed in claim 60, wherein said data traffic controller times the application of its data transmit message and thereby the transmission of data by said second data transmitter onto said expansion bus to ensure that data transmitted from only one of said first and second expansion devices appears on said expansion bus at one time.

62. A method of expanding and operating an expanded interface device, the interface device comprising a first processor with at least one port, an expansion bus adapted to be connected in a data transmitting relationship with selected of a plurality of expansion devices, at least one buffer interconnecting the one port and the expansion bus, said method comprising the steps of:

a) connecting to the expansion device at least first and second expansion devices, the first expansion device includes a second processor;

BEST AVAILABLE COPY

97

b) selecting one of the first and second expansion devices to control the flow of data from each of the first and second expansion devices via the expansion bus to the buffer; and

c) operating the second processor to place a data available message on the one port of the first processor indicating the availability of data destined for the second processor on the buffer.

63. The method of expanding and operating as claimed in claim 62, wherein a second buffer is interconnected between the one port and the expansion bus and there is further included the steps of operating the first processor to respond to the data available message to download the data destined for the first processor and stored in the first buffer to the first processor.

64. The method of expanding and operating as claimed in claim 62, wherein a second buffer is interconnected between the one port and the expansion bus and there is further included the steps of operating the first processor to load data destined for the expansion bus onto the second buffer and to transmit a data available message via the expansion bus to the second processor, and operating the second processor to respond to the data available message to download the data destined for the expansion bus onto the expansion bus.

65. A computer system adapted to be expanded comprising:

a) a first programmable processor including at least one port;

BEST AVAILABLE COPY

98

b) an expansion bus adapted to receive in a data transmitting relationship with selected of a plurality of expansion devices;

c) at least one expansion device that is adapted to be connected in a data transmitting relationship with said expansion bus;

d) at least one buffer interconnected between said one port and said expansion bus; and

e) a second programmable processor included with said one expansion device;

f) one of said first and second programmable processors being multitasked programmed to transmit to the other of said first and second programmable processors a handshake message via said buffer and said expansion bus, said other programmable processor being multitasked programmed to respond to said handshake signal to transmit a signal indicative of the presence of said one expansion device to said one programmable processor confirming that said expansion bus has received said one expansion device.

66. The computer system adapted to be expanded as claimed in claim 65, wherein said one programmable processor is said first programmable processor and said other programmable processor is said second programmable processor.

67. The computer system adapted to be expanded as claimed in claim 65, wherein said one programmable processor is said second programmable processor and said other programmable processor is said first programmable processor.

BEST AVAILABLE COPY

99

68. The computer system adapted to be expanded as claimed in claim 66, wherein there is further included a memory connected to said first programmable processor and said first programmable processor is multitasked programmed to access the content of said memory and to evaluate said presence indicating signal to determine whether said expansion bus has newly received said one expansion device.

69. The computer system adapted to be expanded as claimed in claim 68, wherein said first programmable processor is multitasked to periodically transmit to said second programmable processor said handshake message via said buffer and said expansion bus to said second programmable processor, said second programmable processor multitasked programmed to respond to each handshake message to transmit said presence indicating signal, said first programmable processor being multitasked programmed to access the content of said memory and to evaluate said presence indicating signal to determine whether said expansion bus has previously received said one expansion device to provide an indication that said received one expansion device continues to operate.

70. The computer system adapted to be expanded as claimed in claim 69, wherein said first programmable processor is multitasked programmed to determine whether a presence indicating signal had been previously stored in said memory and, if not, for storing an indication in said memory that said expansion device has received said one expansion device.

71. The computer system adapted to be expanded as claimed

BEST AVAILABLE COPY

100

in claim 70, wherein said first programmable processor is multitasked programmed to determine whether a presence indicating signal had been previously stored in said memory and, if previously stored, for providing an indication that said one received expansion device continues to operate.

72. The computer system adapted to be expanded as claimed in claim 66, wherein said first programmable processor is multitasked programmed to periodically transmit to said second programmable processor said handshake message via said buffer and said expansion bus, said handshake message including a sequence of signals to permit the operation of said second programmable processor to be synchronized with the operation of said first programmable processor.

73. An interface for coordinating the transmission of data between a first data transmitting medium and a second data transmitting medium, the second transmitting medium coupled to a plurality of transponders, each transponder disposed at one of a plurality of sites for measuring the level of a parameter at its site, said interface comprising:

- a) a first controller adapted to be coupled to the first transmitting medium to facilitate communication via the first data transmitting medium with the plurality of transponders;
- b) a second controller adapted to be coupled to the second transmitting medium to facilitate communication via the second data transmitting medium; and
- c) a processor coupled to each of said first and second

BEST AVAILABLE COPY

(b)

processors and programmed to perform at least first and second tasks, said first task is to transmit via said first controller interrogation signals addressed to and prompting each of the plurality of transponders to capture the value of the parameter at its site and to generate and transmit via the first transmission medium return signals bearing respectively the parameter value captured at each site, said second task is to compare each of the received parameter values with a selected limit and if out of limit to generate an indication thereof..

74. The interface as claimed in claim 73, wherein said processor is programmed to perform a third task, said third task is to respond to said out of limit indication to generate an alarm message.

75. The interface as claimed in claim 74, wherein said processor is programmed to perform a fourth task, said fourth task is to transmit said alarm message via the first data transmitting medium to the remote controller.

76. The interface as claimed in claim 73, wherein said interface further includes an alarm coupled to said processor, said alarm responsive to said indication to actuate said alarm.

77. The interface as claimed in claim 74, wherein said processor is programmed to perform a fourth task, said fourth task is to transmit said alarm message via the second transmitting medium to selected of the sites.

78. A system for collecting data from a plurality of

BEST AVAILABLE COPY

(102)

groups of data transponders, each disposed at a site for measuring a value of a parameter at its site, said system comprising:

- a) at least one first interface;
- b) a plurality of second interfaces;
- c) at least one first data transmitting medium and one second data transmitting medium, said second data transmitting medium being coupled to said plurality of second interfaces;
- d) a plurality of third data transmitting media, each of said third data transmitting media being adapted to be coupled to a corresponding group of transponders of said plurality thereof;
- e) a system data base coupled to said first data transmitting media that receives and stores therein said values of parameters from said sites;
- f) said first interface coupled to said first data transmitting medium and to said second data transmitting media and including a first processor is programmed to perform at least a first task, said first task is to provide a plurality of first interrogation signals via said second data transmitting medium to a corresponding one of said plurality of second transponders said third data transmitting medium being adapted to be coupled to the plurality of transponders;
- g) each of said second interfaces of said plurality being coupled to said second data transmitting medium and to a corresponding one of said plurality of third data transmitting media and including a second processor programmed to perform

BEST AVAILABLE COPY

103

at least a second task, said second task is to provide a group of interrogation signals via a corresponding one of said plurality of said third data transmitting media to respective ones of a group of transponders to actuate its transponder to measure the value of its parameters.

79. The system for collecting data as claimed in claim 78, wherein each of said second processors is programmed to perform a third task, each of said second interfaces includes a memory, said third task is to access and store the measured values from the respective transponders of their group in said memory.

80. The system for collecting data as claimed in claim 79, wherein each of said first processors is programmed to perform a fourth task, each of said first interfaces includes a second memories, said fourth task is to access from said first mentioned memories and store the measured values in said second memory.

81. An interface for coordinating the accessing of data from a plurality of transponders and uploading the accessed data via first and second data transmitting media to a remote controller, each transponder disposed at one of a plurality of sites for measuring the level of a parameter at its site, the first data transmitting medium being coupled to the remote controller, the second data transmitting medium being coupled to each of the plurality of transponders, said interface comprising:

a) a first controller adapted to be coupled to the

BEST AVAILABLE COPY

104

first data transmitting medium to facilitate communication via the first data transmitting medium with the remote controller;

b) a second controller adapted to be coupled to the second transmitting medium to facilitate communication via the second data transmitting medium with the plurality of transponders;

c) a memory;

d) a processor coupled to each of said first and second processors and to said memory, said processor is programmed to perform at least first and second tasks, said first task is to access and to store the accessed values of the parameters at its site in said memory, said second task is to respond to an upload command to retrieve from said memory and to transmit a selected part of the values of the parameters via said first controller and the first data transmitting to the remote controller.

82. The interface as claimed in claim 81, wherein said second task is to respond an upload command transmitted by the central controller.

83. The interface as claimed in claim 81, wherein said second task is to respond to an upload command generated at periodic times.

84. The interface as claimed in claim 81, wherein said processor is programmed to perform a third task, said third task is to generate said upload command.

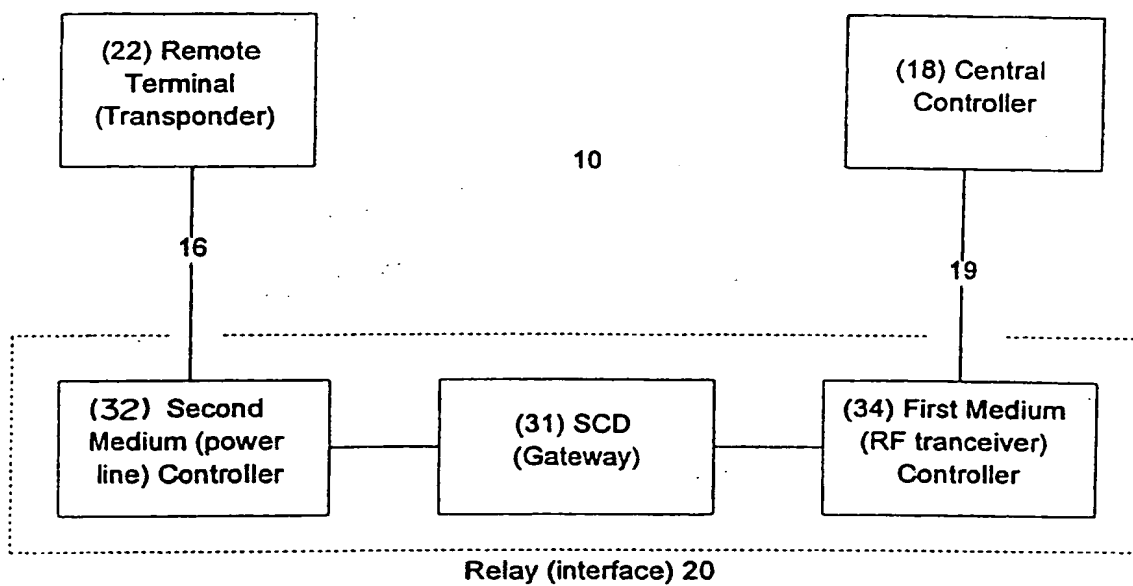
85. The interface as claimed in claim 81, wherein said processor is programmed to perform a third task, said third

BEST AVAILABLE COPY

105

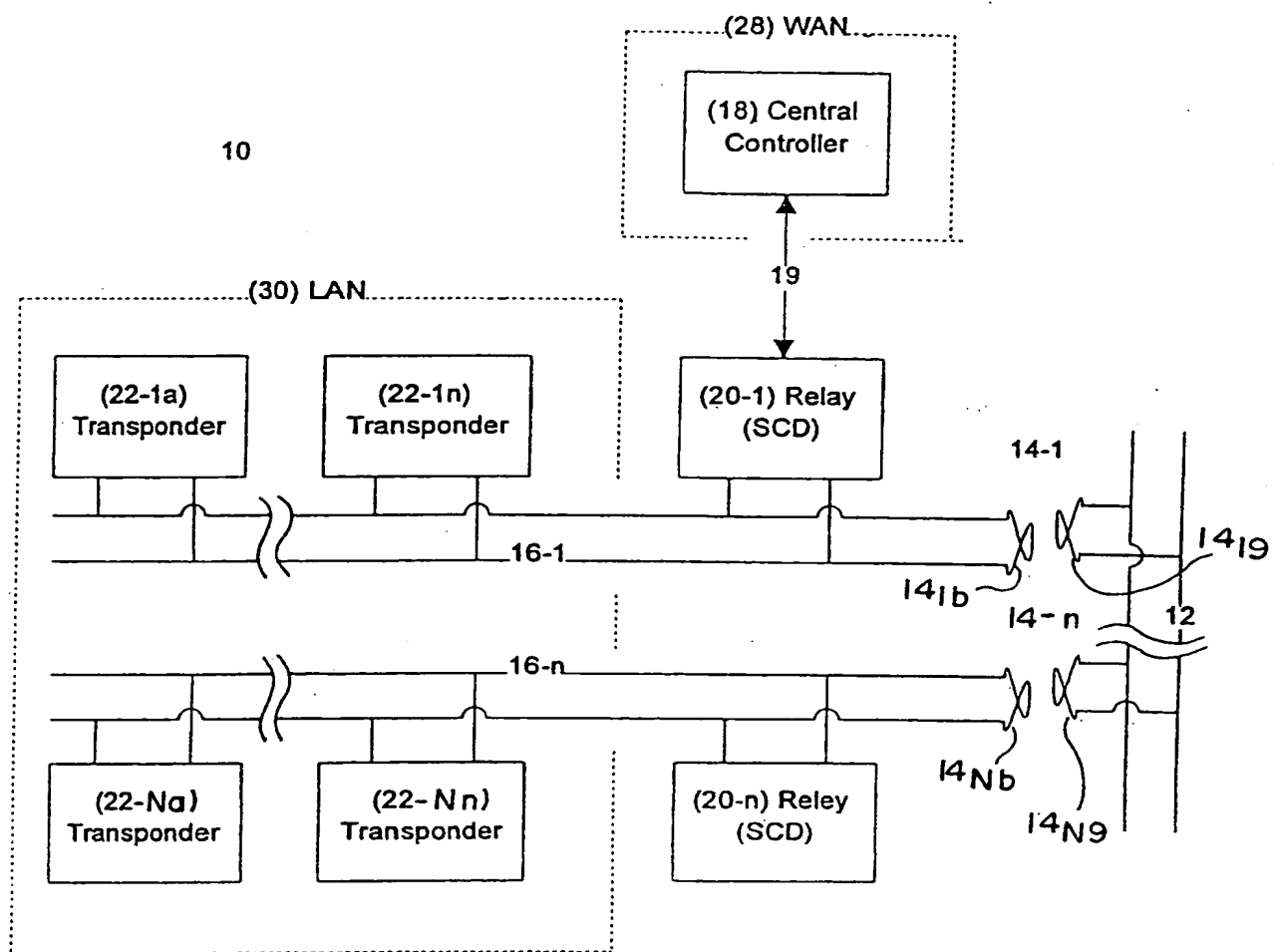
task is to compare each of the accessed parameter values with a selected limit and if out of limit to generate said upload command.

Figure 1



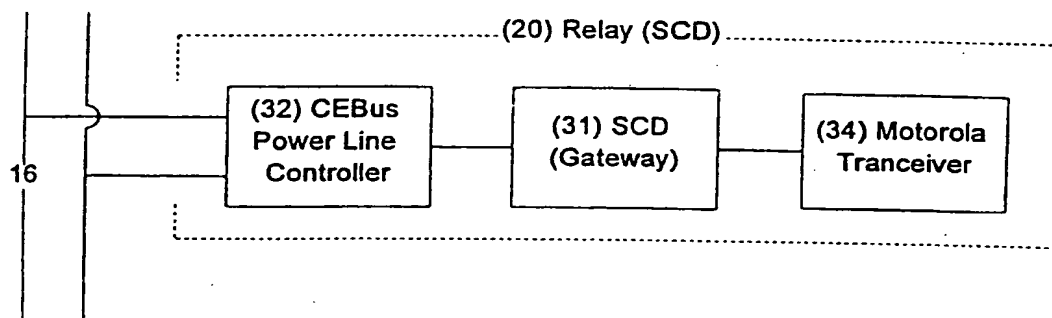
THIS PAGE BLANK (USPTO)

Figure 2



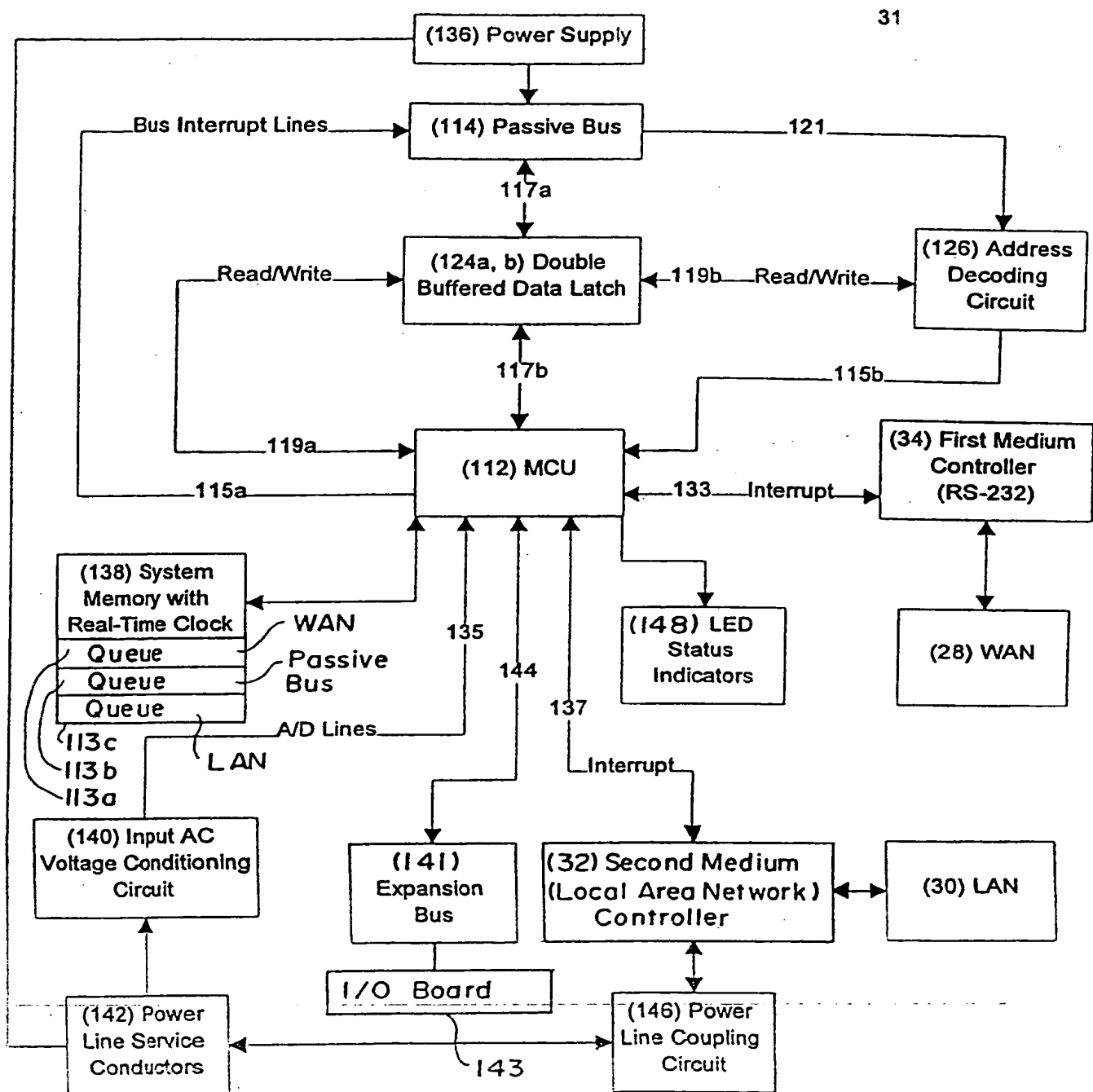
THIS PAGE BLANK (USPTO)

Figure 3



THIS PAGE BLANK (USP 10)

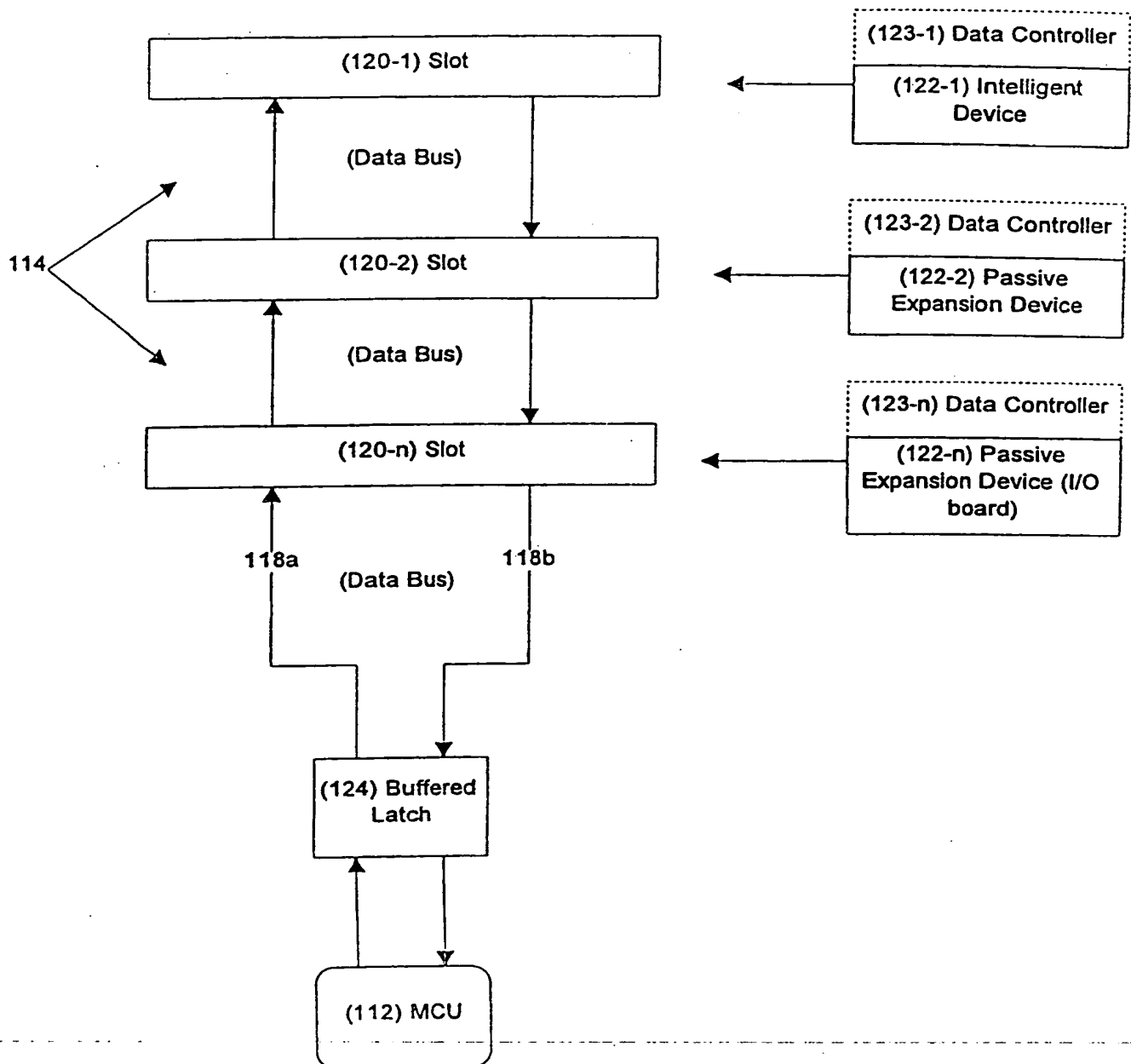
Figure 4



BEST AVAILABLE COPY

THIS PAGE BLANK (USPTO)

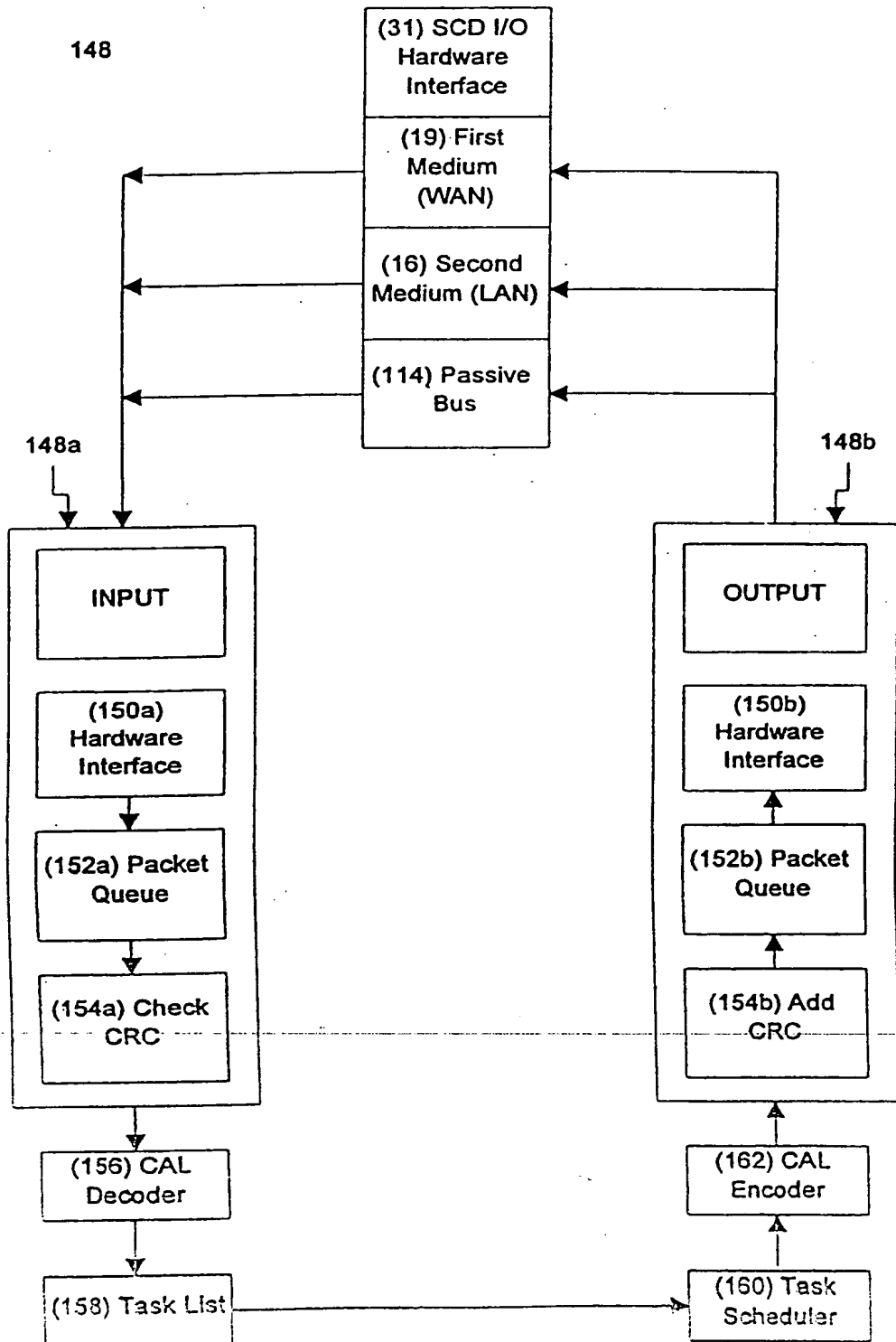
Figure 5



BEST AVAILABLE COPY

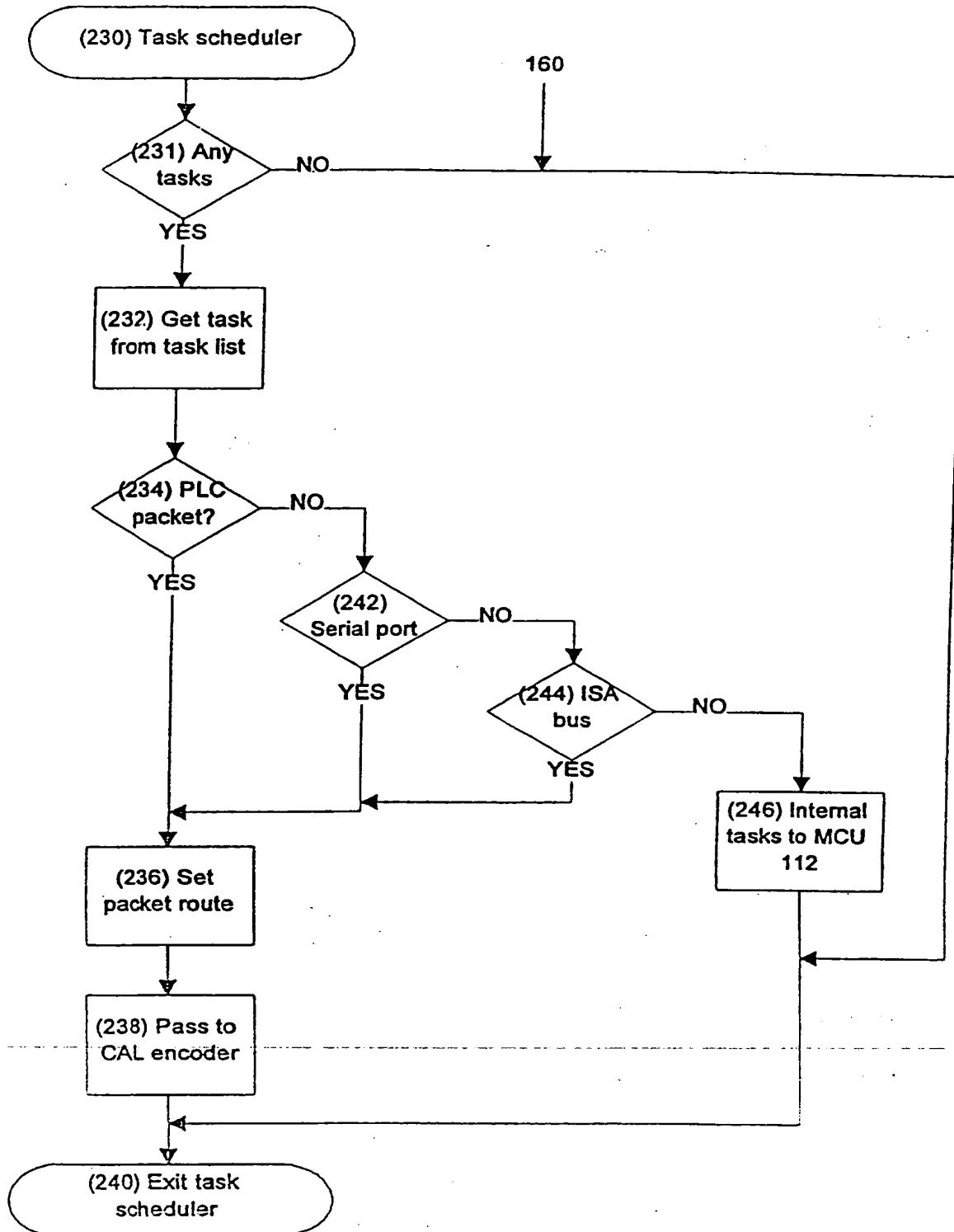
THIS PAGE BLANK (USPTO)

Figure 6



BEST AVAILABLE COPY

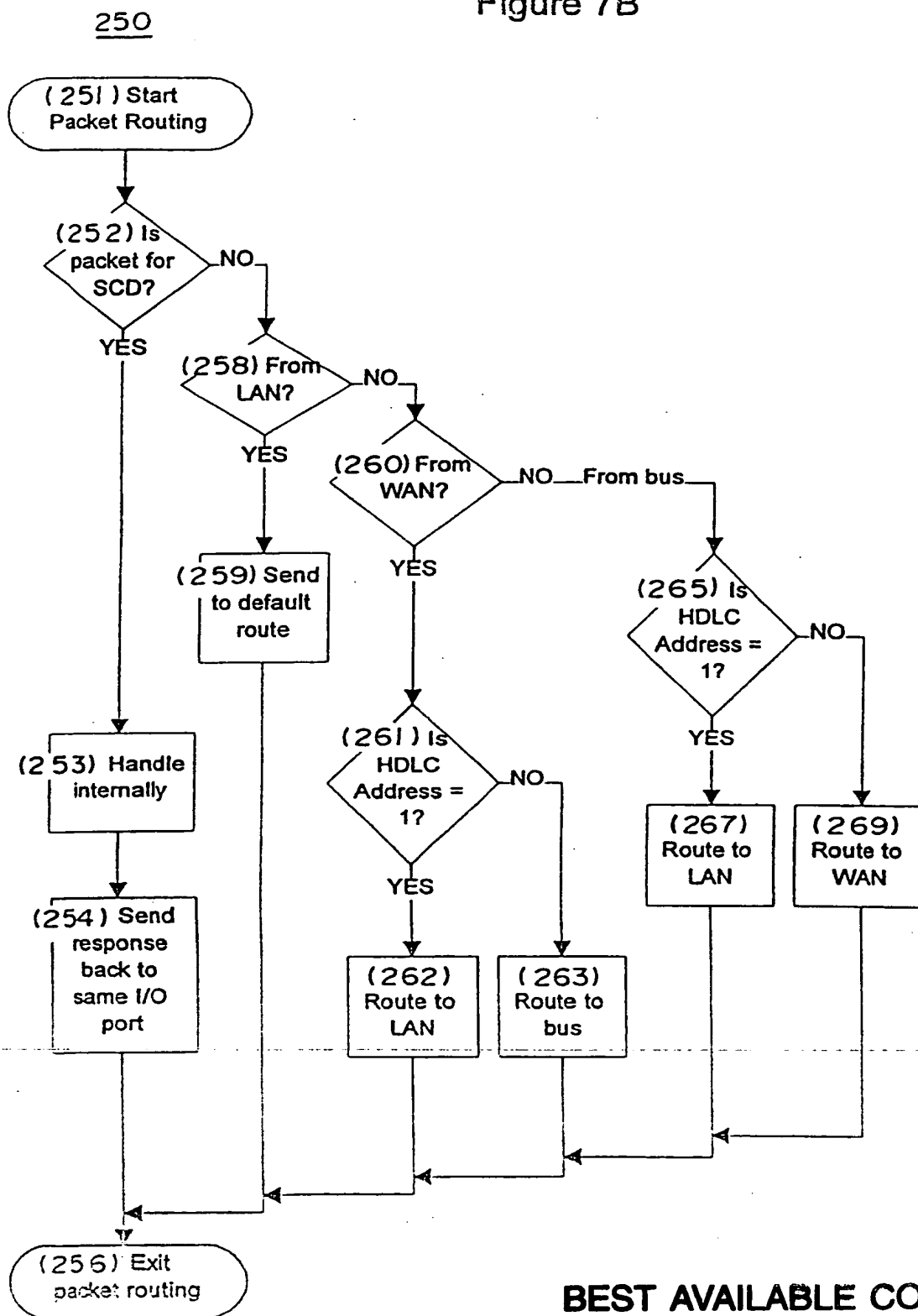
THIS PAGE BLANK (USPTO)
THIS PAGE BLANK (USPTO)

7 / 14
Figure 7A

BEST AVAILABLE COPY

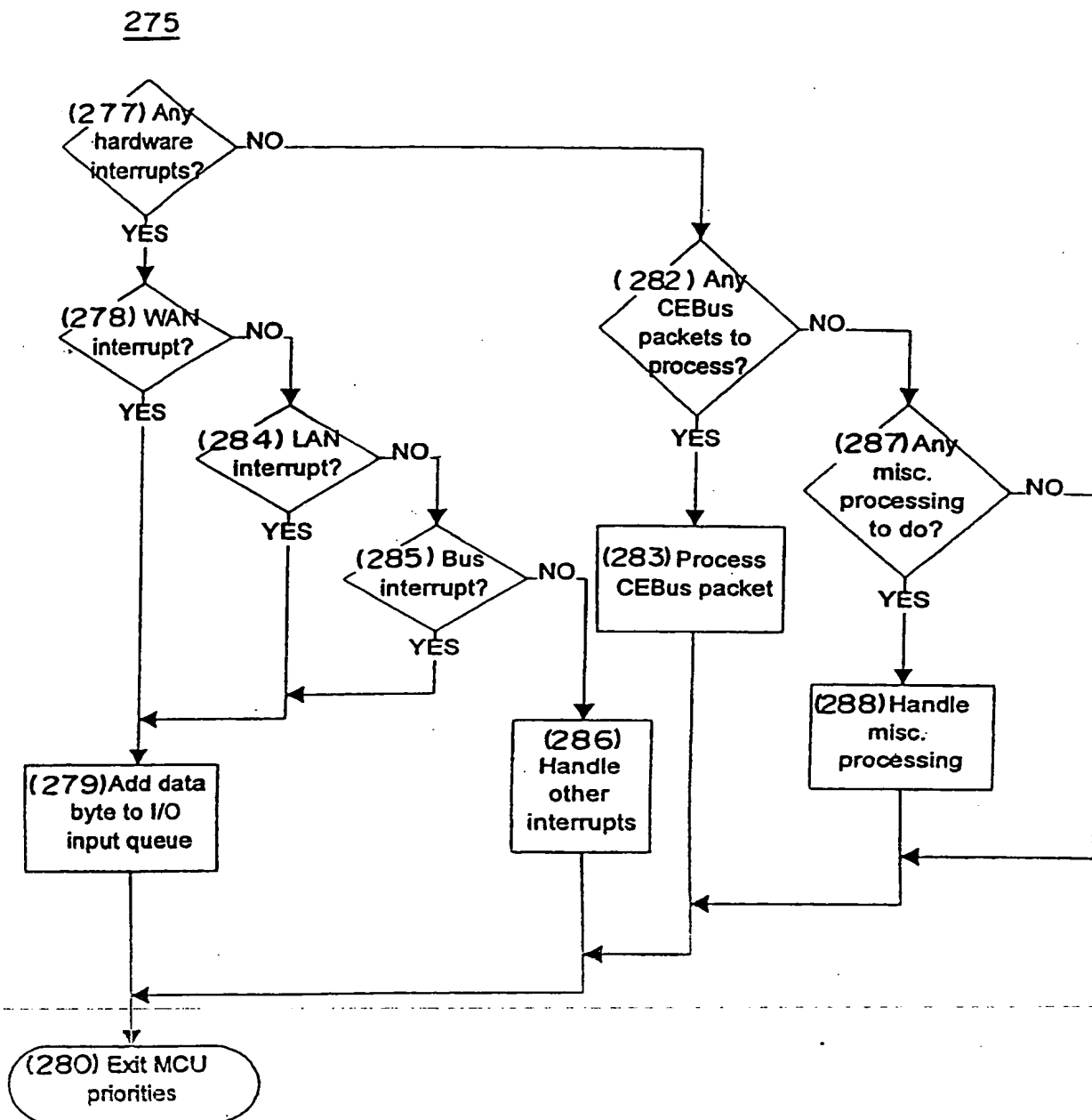
THIS PAGE BLANK (USPTO)

Figure 7B



THIS PAGE BLANK (USPTO)

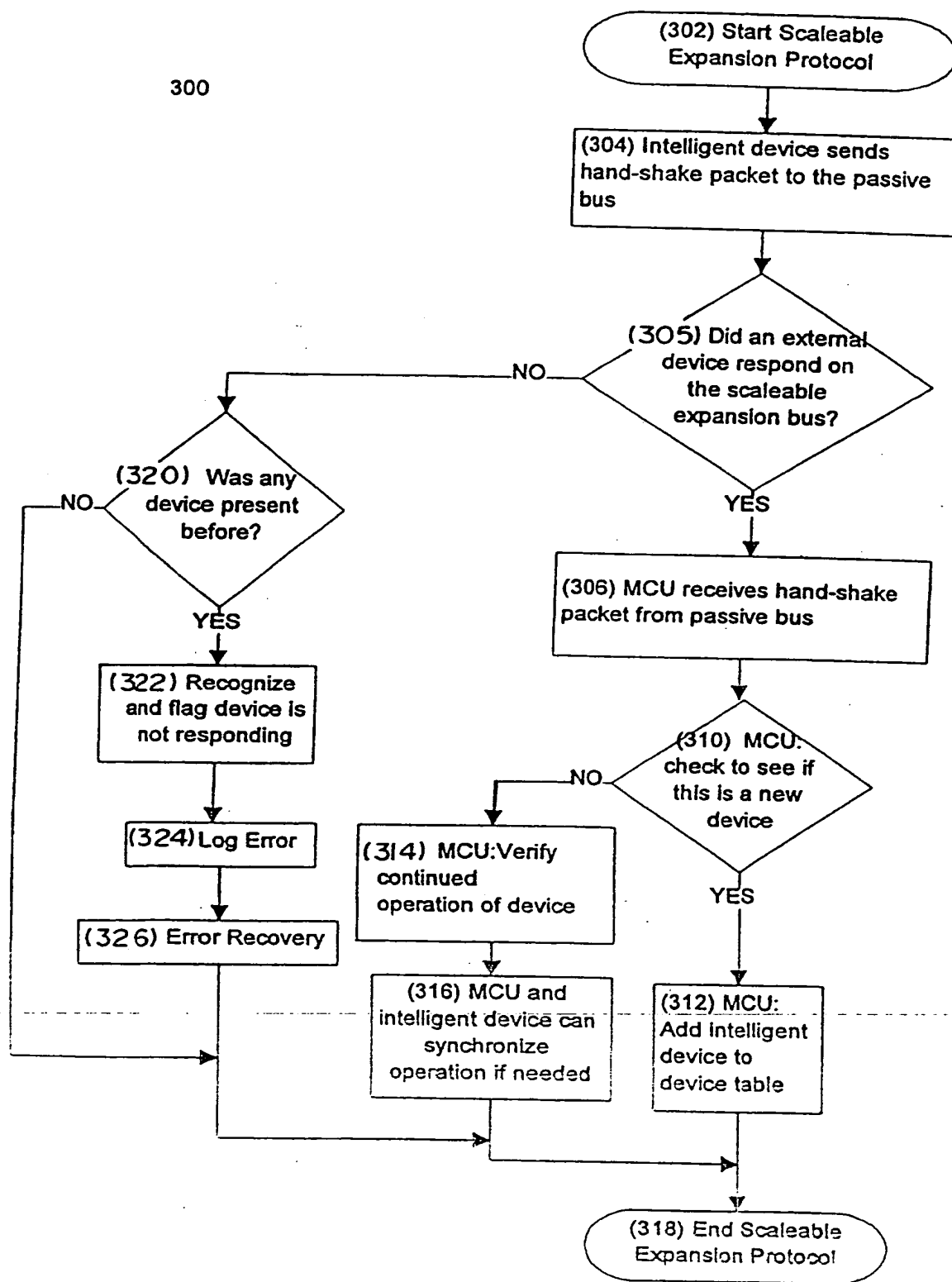
Figure 7C



BEST AVAILABLE COPY

THIS PAGE BLANK (USPTO)

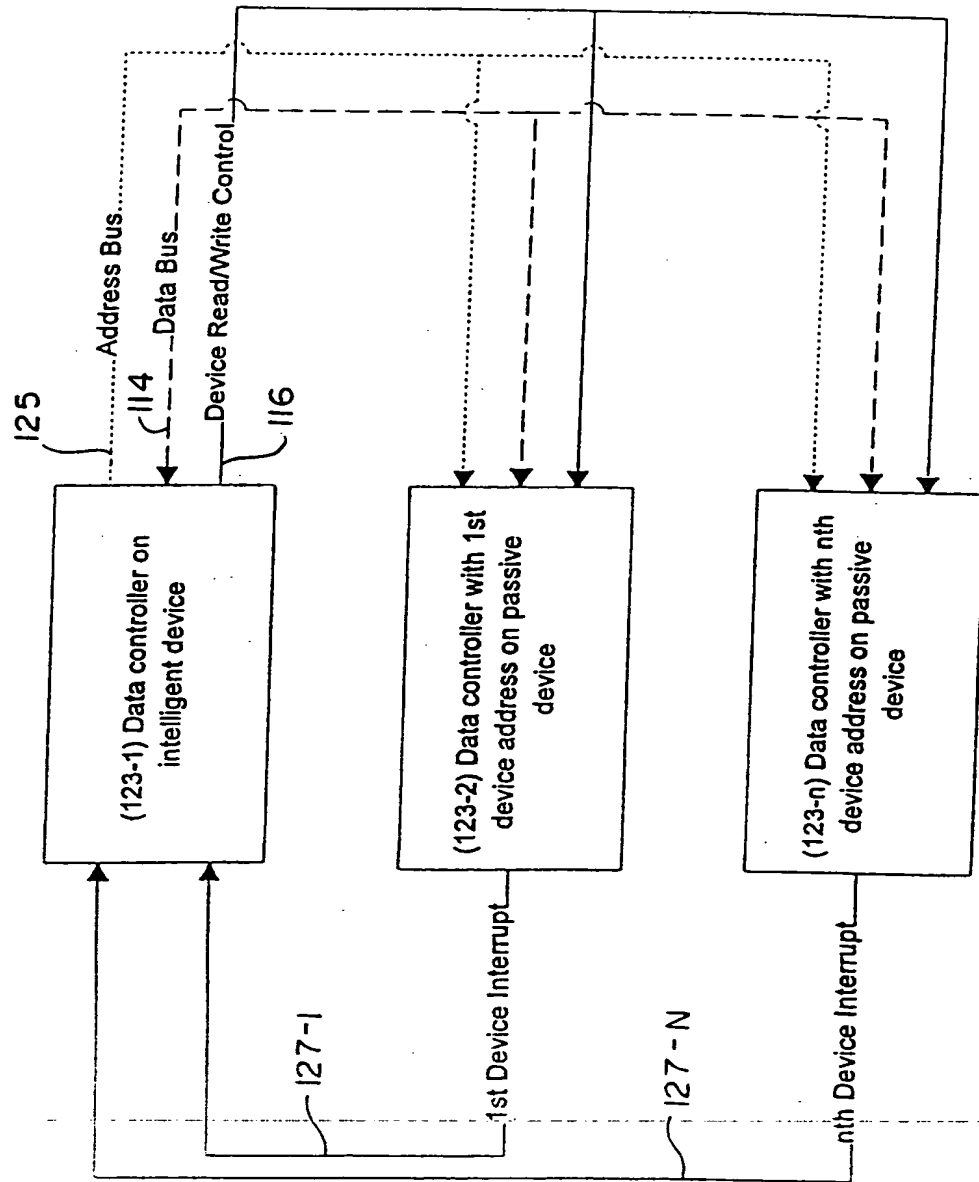
Figure 8A



BEST AVAILABLE COPY

THIS PAGE BLANK (USPTO)

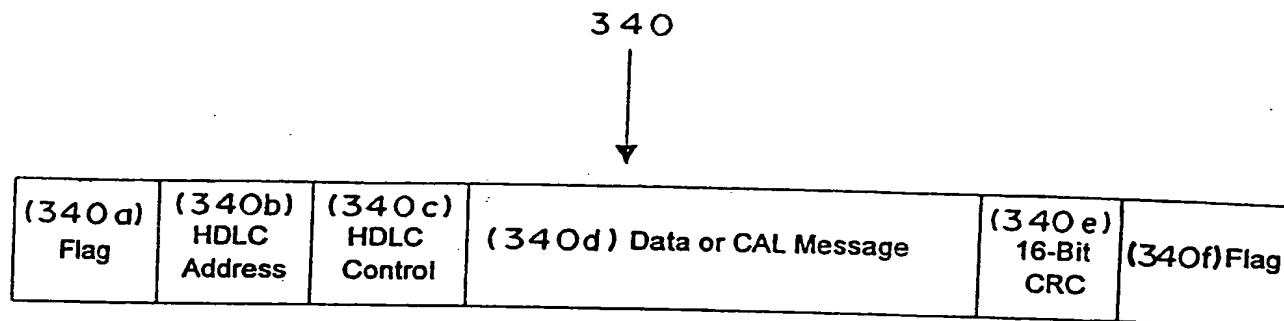
Figure 8B



BEST AVAILABLE COPY

THIS PAGE BLANK (USPTO)

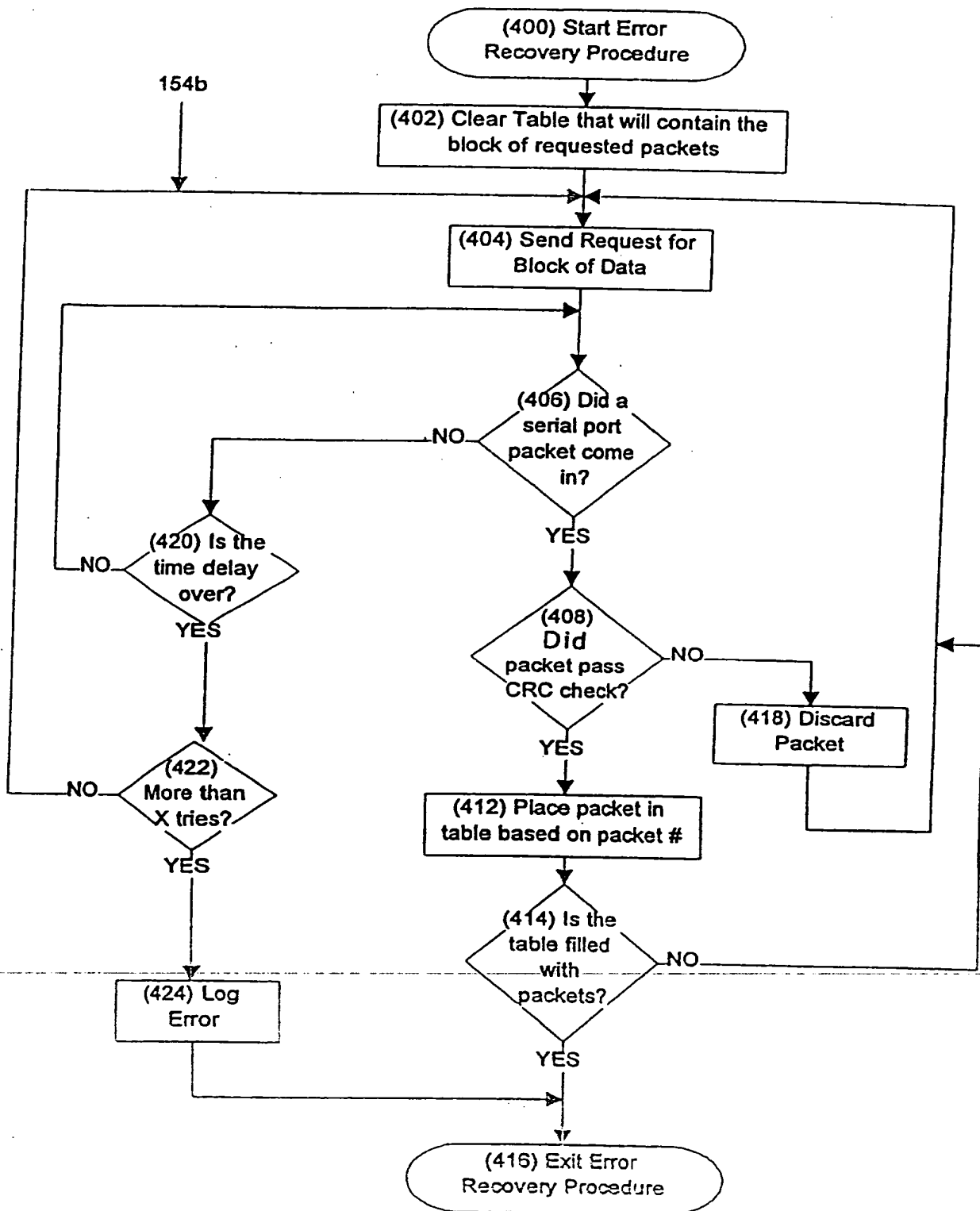
Figure 9



BEST AVAILABLE COPY

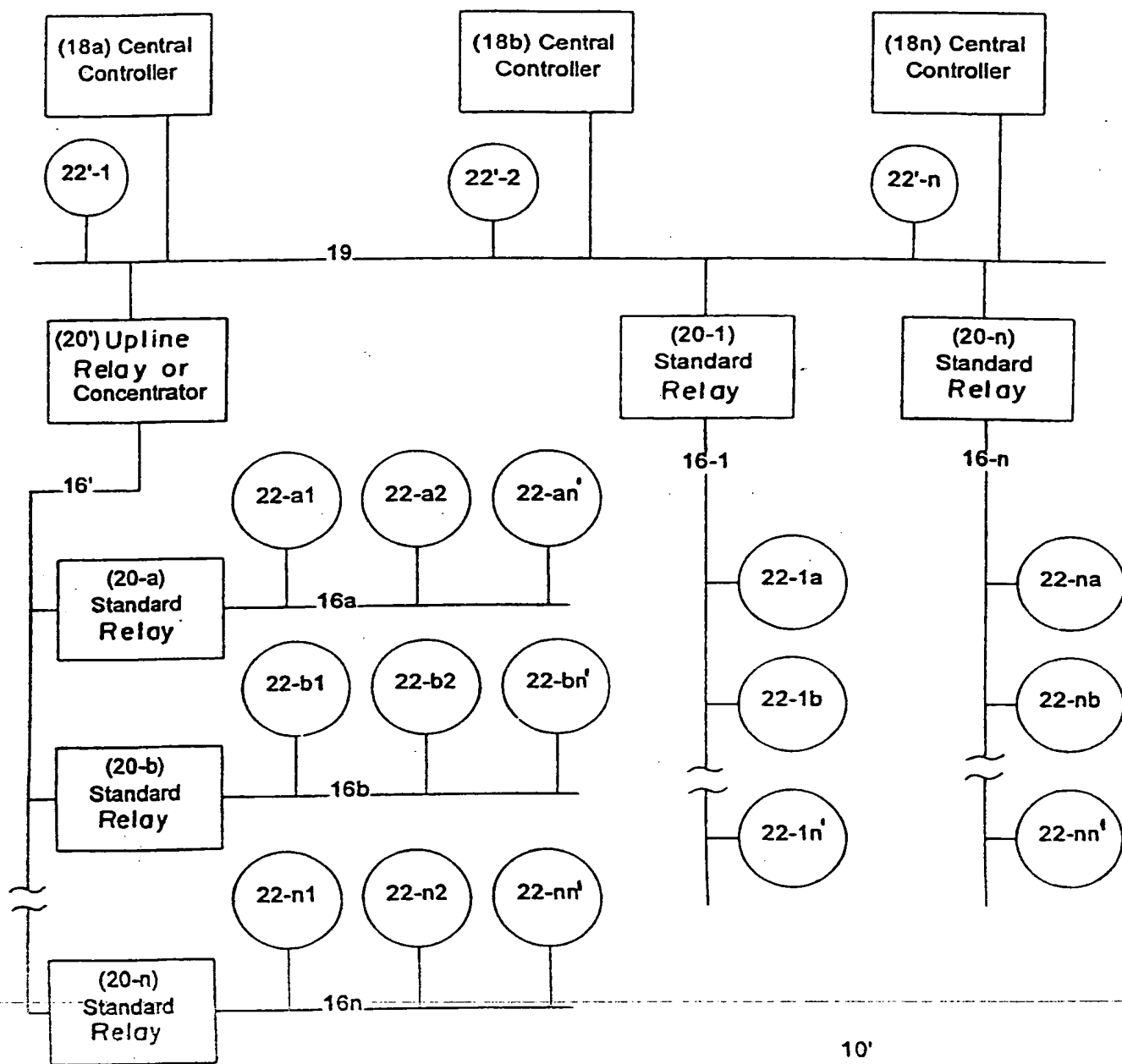
THIS PAGE BLANK (USPTO)

Figure 10



THIS PAGE BLANK (USPTO)

Figure 11



THIS PAGE BLANK (USPTO)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/16734

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) : G06F 13/36

US CL : 395/200.79

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 395/200.79, 200.43, 200.54

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,483,640 (Isfeld et al) 26 February 1993 (26.02.93), FIGS. 1 and 2, col. 4, lines 9-14, 20-25, 58-60, 18-22, 62-64, 11-14, 9-14; col. 5, lines 26-30, 47-51, 42-45.	1-2



Further documents are listed in the continuation of Box C.



See patent family annex.

Special categories of cited documents:	
A document defining the general state of the art which is not considered to be of particular relevance	*T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
E earlier document published on or after the international filing date	*X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
O document referring to an oral disclosure, use, exhibition or other means	*Z* document member of the same patent family
P document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search

02 OCTOBER 1998

Date of mailing of the international search report

23 OCT 1998

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

Glendon Burgess

Telephone No. (703) 305-3900

THIS PAGE BLANK (USPTO)
